Energy efficiency in wireless ad hoc and sensor networks: routing, node activity scheduling and cross-layering

Saoucene Mahfoudh

THESE DE DOCTORAT
DE L'UNIVERSITÉ PARIS 6 – PIERRE ET MARIE CURIE

to obtain the degree of

Docteur de l'Université Paris 6

Discipline: Computer Science

Host laboratory: INRIA Rocquencourt

by

Saoucene Mahfoudh

Title:

Energy efficiency in wireless ad hoc and sensor networks: routing, node activity scheduling and cross-layering

Thesis Director: Pascale Minet

Jury:

Reviewers: M. Mario Gerla
M. David Simplot-Ryl

Examiners: M. Tuan Dang
M. Philippe Jacquet
Ms. Pacale Minet
M. Michel Misson
M. Guy Pujolle
Ms. Leila Saidane

To my parents: Alaya Mahfoudh and Samira Hassouna
To my husband and son: Walid and Hamza Ridene

Acknowledgements

It is with great pleasure and felicity that I would like to express thanks to all people who have made this thesis possible.

First of all, I want to express my deep gratitude to Ms. Pascale Minet, my advisor, for her excellent support, her advice and guidance during this research. Her constant availability, academic rigor and experience helped me to develop excellent research habits and gave me the will and the power to reach success. I can not thank her enough for her direction.

I would like to thank Mr. Philippe Jacquet for providing me the opportunity to be a member of his team.

My cordial thanks go to Mr. Mario Gerla and Mr. David Simplot-Ryl for accepting to review my thesis. I would also like to thank Mr. Tuan Dang, Mr. Philippe Jacquet, Mr. Michel Misson, Mr. Guy Pujolle and Ms. Leila Saidane who honored me by accepting to be part of my thesis committee.

I would like to express my gratitude to all the member of Hipercom team for the exceptionally friendly working environment. Thanks to Christine Anocq for her availability and help in administratif tasks. Thanks to Amina, Cedric, Skandar, Ichrak, Anis, Paul, Yasser, Yassine and Salman. Working with you was a great pleasure.

A special thank goes to my husband, Walid Ridene for his support, understanding and encouragement throughout all my thesis and all the happiness and the love that he brings to my life.

Last but not least, I want to thank my parents, Alaya Mahfoudh and Samira Hassouna. Their faith in me has encouraged me to go after my dreams. Thanks to my sister Salma, my brothers Slim, Mohamed, Houssem and all my family, especially Khaled and Olfa, for their constant motivation. My love and reverence for them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Wireless networks play a crucial role in the communication systems nowadays. Wireless networks are being increasingly used in the communication among devices of the most varied types and sizes. User mobility, affordable-ity, flexibility and ease of use are few of many reasons for making them very appealing to new applications and more users everyday.

In this work, we consider only wireless networks capable of operating without the support of any fixed infrastructure. We also consider the general case of multi-hop networks. More precisely, we will consider wireless ad hoc networks as well as wireless sensor networks.

The diversity of the applications supported by wireless ad hoc and sensor networks explain the success of this type of network. These applications concern as various domains as environmental monitoring, wildlife protection, emergency rescue, home monitoring, target tracking, exploration mission in hostile environments, etc. However, the most critical requirement for adopting such networks is energy efficiency. Indeed, some nodes are battery operated and battery replacement can be difficult, expensive or even impossible. The goal of communication protocol designers is then to maximize the lifetime of such networks.

1.1.1 Specificities of wireless ad hoc and sensor networks

Wireless ad hoc and sensor networks have in common some characteristics that have to be taken into account by energy efficient techniques:

- **Lack of pre-configuration**: A wireless ad hoc and sensor network is a collection of wireless nodes that can dynamically self-organize into an arbitrary and temporary topology to form a network without using any pre-existing infrastructure.
- **Wireless communication**: which has the following properties:

- Radio interferences: Indeed, when a node N_1 is transmitting to a neighbor node N_2, no other node in the transmission range of N_1 can receive another frame. Similarly, no other node in the transmission range of N_2 can send another frame.
- Radio link versatility: as the propagation conditions change very frequently, the quality of a radio link varies strongly in the time.
- Limited bandwidth: the wireless bandwidth has a capacity much smaller than a wired one due to the shared nature of the wireless channel and interferences.

- **Scalability**: Communication protocols should be able to support large (i.e. a high number of nodes) or dense (i.e. a high number of neighbors per node) wireless networks.

1.1.2 Differences between wireless ad hoc and sensor networks

While wireless sensor networks share many commonalities with existing ad hoc network concepts, there are also a number of differences and specific challenges:

- **Applications.** The main difference between common ad hoc networks and wireless sensor networks is their different area of application. In fact, a sensor network is characterized by its strong interaction with its environment. For this reason, it can be used in a large number of applications [1] like:
  - Indoor/outdoor environment monitoring,
  - Monitoring of buildings/bridges/airplanes for structural integrity,
  - Ubiquitous computing for healthcare applications,
  - Sensing within factory environments,
  - Sensing for transportation applications,
  - Warehouse and retail inventory management,
  - Interactive museums/exhibits, etc.
- **Constraints.** Constraints in wireless sensor networks are stronger than in ad hoc networks because of the miniaturization of sensor devices:
  - Limited memory/storage.
  - Limited processing capacity.
  - Radio transceiver: the radio range is generally shorter (20m in 802.15.4 versus 250m in 802.11b).
  - Low rate: 250 kbps in 802.15.4 versus 11Mbps in 802.11b.
  - Limited energy: The energy constraint in wireless sensor networks is stronger than in ad hoc networks. This is because, sensor networks are usually deployed in hostile environment or the number of nodes in the network can be very important. Consequently, changing the nodes batteries can be very difficult, expensive or even impossible.

- **Mobility support**: It is an evident requirement for VANETs (Vehicular Ad hoc Networks), for instance. It is usually not required in many wireless sensor networks: in these networks, either all nodes are fixed or a very limited number of them is allowed to move.

1.2 Problem Statement

In wireless ad hoc and sensor networks, nodes working only with battery power will die after battery exhaustion. This means that the network has a limited lifetime. One of the main challenges facing the network designers in wireless ad hoc and sensor networks is to maximize the network lifetime. Our work is centered on energy efficient techniques that will prolong network lifetime. Several classes of energy efficient techniques exist. Among them topology control assumes that the MAC layer is able to tune the transmission power of the node. In this work, we do not make such an assumption: the sender always transmits with the maximum transmission power. In the following, we focus on the three other energy efficient classes: energy efficient routing, node activity scheduling and optimization of the volume of data transferred.

The specificities of the wireless ad hoc and sensor networks make difficult the design of communication protocols and in particular energy efficient ones. They also lead to favor simple techniques: protocols should induce a low overhead, have minimal memory and computation requirements, etc.

1.3 Contributions

The main contribution of this work are:

- **Design, specification and performance evaluation of an energy efficient routing protocol based on OLSR.** The proposed protocol, called EOLSR, computes the path built from intermediate nodes with sufficient residual energy that minimizes the energy dissipated by an end-to-end transmission. The second originality of this protocol resides in the cross layering with both the application and the MAC layer. The cross layering with application allows us to define two modes depending on the application requirements:
  - Generic mode applies to applications where the communications characteristics are not known in advance.
  - Strategic mode applies to applications where the potential destinations exist in a limited number and are known in advance. This is the case for instance in data gathering applications.
- **Design, specification and performance evaluation of node activity scheduling.** The proposed solution, named SERENA, is based on node coloring to assign time slot to nodes.

The innovation resides in taking into account wireless environment constraints like unidirectional links, immediate acknowledgement of unicast transmissions, etc. The second innovation consists in adapting to application requirements by selecting the most appropriate mode:

  – Generic mode is used where there is no specific knowledge on the application and topology.
  – Specific mode is used in applications organized in a tree, like data gathering applications.

- **Benefits brought by the cross layering.** In this work, we have advertised a cross layering approach to optimize communication protocols taking into account the real environment in which the network operates. We have shown that high benefits can be obtained at a small cost. For instance, we have specified the interactions between:

  – the routing protocol and both the MAC and application layers to optimize the stability of routes as well as reduce the overhead.
  – node activity scheduling and both the MAC and application layers to detect color conflicts and to better adapt to application requirements.

- **Specification and performance evaluation of optimized broadcast.** We have shown that the solution proposed, based on multipoint relay, improves network lifetime.

- **Integration in the network layer of the OCARI project.** We have contributed to the specification of the energy efficient network layer of the OCARI project. We have specified network messages format, network primitives and processing done by the network layer. This network layer integrates EOLSR, SERENA, and optimized broadcasts. It takes advantages of cross layering with the MAC and application layers.

1.4 Thesis organization

This dissertation is structured as follows:

- **Chapter 1, this chapter, defines the context and states the problem** that will be solved in this work. It points out the main contributions of this thesis and presents the organization of this dissertation.

- **Chapter 2 deals with energy efficiency**. It first defines the main concepts such as network lifetime and node energy consumption model. It then provides a classification of energy efficient techniques and gives for each of them a brief state of the art. Simulation results show the distribution of node energy consumption in wireless sensor networks.

- **Chapter 3 focuses on energy efficient routing**. First, we recall some work related to multipath routing and minimum energy routing. We show that OLSR does not maximize the network lifetime and present EOLSR, our energy efficient extension of OLSR. We define the four main modules of EOLSR: energy consumption model, energy efficient selection of multipoint relays, routing algorithm and optimized broadcasts. Performance evaluation results allow us to validate the design choices made for each module. A cross layering optimization with the application layer leads us to distinguish between a generic mode adapted to applications where the communications are not known in advance and a strategic mode where all the destinations are known in advance. The strategic mode is well adapted to data gathering applications, for instance.

- **Chapter 4 is centered on node activity scheduling**. We are interested more particularly in graph coloring. We first compare the respective merits of vertex and edge colorings and present the benefits of each coloring in wireless sensor networks. We precisely define the problem of node coloring in these networks highlighting the different constraints. SERENA's two-hop coloring algorithms is described with all exchanged messages and the information maintained by each node. The coloring algorithm is then followed by a slot assignment algorithm. Performance evaluation of SERENA is done, comparing the results of SERENA with other coloring algorithm and other variations of TDMA. SERENA is then extended to allow immediate acknowledgement of unicast transmissions and to cope with unidirectional links that exist in a real environment. Color conflicts cannot be avoided in an environment prone to topology changes due to unidirectional links, node mobility and late node arrivals. Such conflicts, detected by a cross layering approach with the MAC layer, are solved by SERENA.

- **Chapter 5 shows how to optimize node activity scheduling for data gathering applications** where the delay needed to detect data from all sensor nodes must be minimized. Simulation results allow us to compare SERENA with classical TDMA and TDMA-ASAP a dynamic variant of TDMA. Network calculus is used to compare the performance of our algorithm with ZigBee in cluster tree topology.

- **Chapter 6 deals with cross layering and the integration of energy efficient techniques**. After a brief presentation of different cross layering architectures, we select the one adopted in this work to support the cross layering for SERENA and for EOLSR. For each of them, we detail the cross layering with (1) the application layer leading to distinguish two modes according to application requirements and (2) the MAC layer. Finally, we describe the OCARI project for which we are in charge of specifying an energy efficient network layer integrating EOLSR and SERENA. We also show how to generalize our coloring algorithm to any set of application requirements.

- **Chapter 7 concludes this dissertation and provides directions for further researches.**

Chapter 2

Energy efficiency: State of the art

2.1 Introduction

The diversity of the application supported by wireless ad hoc and sensor networks explain the success of these networks. However, nodes in such networks can have a limited amount of energy. This energy can be very expensive, difficult or even impossible to renew. So saving energy to maximize network lifetime is one of the critical problems in wireless ad hoc and sensor networks. That is why algorithms and protocols operating in such networks should be energy efficient. Several solutions are proposed to improve the network lifetime.

In this chapter, we start by giving different definitions of the network lifetime. Then, we present the energy consumption model the most frequently used. We introduce the different states that a wireless node can take and the different reasons of energy consumption in the network. In next section, we give a review of energy efficient solutions proposed for wireless ad hoc and sensor networks. We classify these solutions into four classes. We detail each class and give some examples. Then, we present some solutions that adopt cross layering to be energy efficient. Finally, we analyze the node energy consumption distribution in wireless ad hoc and sensor networks and identify the states in which nodes consume the highest part of their energy.

2.2 Definition of network lifetime

All energy efficient techniques share the same goal: to maximize network lifetime. Unfortunately, there is no definition of network lifetime commonly agreed in the literature.
Several definitions of network lifetime exist, the most frequently used are:

- Definition D1: Time to first node failure due to battery outage. As sensor redundancy is generally used, a sensor failure can have no influence on the network and application functionalities. That is why, some authors prefer the time to the failure of a certain percentage of the sensors (e.g. 20%), in order to take into account possible redundancy.

- Definition D2: Time to application failure: an application functionality is no longer ensured (e.g. a vital parameter of a patient is no longer monitored). Definition D1 differs from definition D2 because of redundancy in sensor coverage. Indeed, if an area is covered by k sensors, the failure of k-1 of them is perfectly tolerated.

- Definition D3: Time to first network partitioning. As soon as the network is no longer connected, vital information can no longer be transferred to its destination.

In the absence of knowledge of the application supported by the network, definitions D1 and D3 are the most useful ones to compare different energy efficient strategies. In all cases, an energy consumption model is needed to conclude in favor of an increase in network lifetime. In next section, we present the energy consumption model the most frequently used.

2.3 Energy consumption model

There are many energy consumption models proposed in the literature. We can unified [25] these models by the following one highlighting two components of the energy dissipated by the transmitter. The first component reflects the energy consumed by the radio. The second component presents the energy consumed by the amplifier and depends on the distance between the transmitter and the receiver.

$$E_{transmit} = C_1(size) + C_2(size, d) \;\; = C_1 \cdot size + C_2 \cdot size \cdot d^{\alpha} \;\; = size(C_1 + C_2 \cdot d^{\alpha}), \qquad (2.1)$$

with: C_1: Energy consumed by the radio of the transmitter to transmit a bit,
C_2: Energy consumed by the amplifier to send a bit at a distance of 1 meter,
$size$: Packet size,
d: Distance between the transmitter and the receiver,
and $0 < \alpha < 6$ values of 2 or 4 are the most frequently used.
Many works about topology control focus on the component proportional to the distance. Equation 2.1 becomes, when uniformed by the size of the transmitted packet:

$$E_{transmit} = C_1 + C_2 \cdot d^{\alpha}.$$

This formula points out the relation between energy consumption and distance. This relation is used in topology control to optimize energy consumption by tuning the transmission power taking into account the distance between the transmitter and the receiver.
Many other works suppose that the transmissions is done at the maximum power. In other words, the transmitter uses the transmission power such that any receiver at a distance equal to the transmission rage correctly receives the message. Consequently, we can consider the quantity $(C_1 + C_2 \cdot d^{\alpha})$ as a constant named C. Hence, the energy dissipated in a transmission by a transmitter is : $E_{transmit} = C \cdot size$ where $size$ denotes the packet size in bits.
In our work, we assume that: the transmission power is a constant and the same for all nodes in the

network.
Concerning the receiver, it consumes energy to capture packets. this energy is expressed as follows:

$$E_{receive} = C_1(size) = C_1 \cdot size, \tag{2.2}$$

with: C_1: Energy consumed by the radio of the receiver to capture one bit,
$size$: Packet size.
We notice that this energy can not be neglected in the energy dissipated during transmissions.

2.4 Energy consumption in wireless networks

In addition to the transmit and receive states a wireless node can be in the idle or sleep states. In next section we present and explain each wireless node state.

2.4.1 Energy states

A wireless node can be in one of the following four states: Transmit, Receive, Idle or Sleep. Each state corresponds to a different power level (see Table 2.1):

- **Transmit**: node is transmitting a frame with transmission power $P_{transmit}$;
- **Receive**: node is receiving a frame with reception power $P_{receive}$. This frame can be decoded by this node or not, it can be intended to this node or not;
- **Idle** (listening): even when no messages are being transmitted over the medium, the nodes stay idle and keep listening the medium with P_{idle};
- **Sleep**: the radio is turned off, and the node is not capable of detecting signals: no communication is possible. The node uses P_{sleep} that is largely smaller than in any other state: the energy consumption is minimum;

State	Power value (Watt)		Current (mA)
	802.11	802.15.4	802.15.4
Transmit	1.3	0.1404	33.1
Receive	0.9	0.1404	33.5
Idle	0.74	-	-
Sleep	0.047	0.000018	0.005

Table 2.1: Power value in each radio state.

In Table 2.1, we report the reference values of power consumption in each state taken from a Lucent silver wavelan PC card [58] implementing the IEEE 802.11b medium access [14] and a ZigBee node [59] implementing IEEE 802.15.4 [60] medium access. In both casees, we can notice that the least consuming state is the *sleep* state. However, the power used in transmit and receive state is close for the IEEE 802.15.4 medium access.

2.4.2 Reasons of energy consumption in the network

In wireless ad hoc and sensor networks, nodes dissipate energy in processing, transmitting and receiving messages. This energy is needed for correct working of the wireless networks. In addition to this energy, there is a great amount of energy wasted in states that are useless from the application point of view, such as:

- idle listening: since a node does not know when it will receive a message it must permanently listen to the medium and so it remains in the idle state. As we can notice in Table 2.1 the power used in idle state is close to the power used in receive state.
- overhearing: When a sender transmits one packet to next hop, because of the shared nature of wireless medium, all neighbors of the source receive this packet even if it is intended to only one of them. Thus the overhearing is the energy dissipated when the node is an one-hop neighbor of the sender and is not the destination.
- interference: Each node situated between transmitter range and interference range receives this packet but it cannot decode it.
- collision: In case of CSMA/CA medium access, When a collision occurs, the energy dissipated for the transmission and for the reception of colliding frames is wasted.

The energy constrained nature of wireless nodes requires the use of energy efficient strategies to minimize the energy wasted in these useless states and so maximize network lifetime. In the next section, we describe and classify works aimed at minimizing energy consumption and improving network lifetime.

2.5 Classification of energy efficient techniques

With the energy constrained nature of wireless networks, it is very important to use energy efficiently. Many researches address this issue in wireless networks. their goal is to maximize the network lifetime. We can classify these researches in four strategies.

1. **Energy efficient routing**: the goal is to minimize the energy consumed by the end-to-end transmission of a packet, to avoid nodes with a low residual energy and reduce the number of unsuccessful transmissions, like [27], [28], [29], [30], [31], [32], [33];

2. **Node activity scheduling**: the idea of scheduling node activity is to alternate node states between sleeping and active to minimize energy consumption while ensuring the network and application functionalities, like [7], [8], [9];

3. **Topology control by tuning node transmission power**: these strategies find the optimum node transmission power that minimizes energy consumption, while keeping network connectivity, like for instance [2, 3];

4. **Reducing the volume of information transferred**: These strategies consist in:

    - aggregating information with the use of clusters, like [25, 26] or without, like [4], [5], [6];
    - decreasing the frequency of information refreshment with distance, like [45];
    - avoiding to transfer information to uninterested nodes.
    - optimizing the broadcast of messages throughout the network.

We now detail each of these four classes of energy efficient strategies.

2.5.1 Energy efficient routing

There exist many energy-aware routing protocols designed for wireless sensor networks. Our purpose is not to give an exhaustive list but rather to present a classification of these protocols. All of them aim at minimizing the energy consumed either in active communications, or in inactivity periods, or both. Some of them make assumptions on the application model (e.g. query of data meeting some attributes). Some build clusters to reduce the number of transmissions. Others use location based information to limit the broadcast of queries. Furthermore, some of them use multipath routing, in order to balance the energy consumption over the network nodes. They can also be distinguished according to the criteria used to select routes. Finally, we get the classification [15] of Table 2.2 and organized according to the following criteria: data centric, hierarchical, geographical, energy criteria used for route selection, multipath routing, support of sleeping nodes.

In this table residual means that the residual energy of nodes is taken into account. Overhear means that the route minimizing the energy dissipated during a transmission is selected, taking into account the energy lost in overhearing. Interference means that the route minimizing the energy dissipated during a transmission is selected, taking into account the energy lost in interferences.

A data centric routing protocol optimizes traffic by querying sensor nodes based on their data attributes or interests. It assumes a query driven model of data delivery and a routing based on data attributes. A generic routing protocol does not make such assumptions. Examples of data centric routing protocols are SPIN [6], and Directed Diffusion [16]. In SPIN, a node advertises the availability of data allowing interested nodes to query that data. However, this protocol does not ensure the end-to-end delivery of data if intermediate nodes are not interested in that data. In Directed Diffusion, the sink broadcasts an interest message to sensors, interested sensors reply with

Protocol	Data centric	Hierarchical	Geographical	MultiPath	Route selection	Sleep mode
SPIN [6]	yes					
Directed Diffusion [16]	yes					
LEACH [25]		yes			residual	
TEEN [17]	yes	yes				
EAP [24]		yes				yes
GAF [20]			yes			yes
GEAR [19]			yes			
REAR [32]					residual	
Ganesan [28]				yes		
Srinivas [29]				yes		
Shah [27]				yes	residual	
Kwon [31]					residual overhear interference	
EOLSR [**?**]					residual overhear interference	yes with SERENA

Table 2.2: Classification of energy aware routing protocols [15].

a gradient message. Both interest and gradient messages allow the sink to establish paths to sensor nodes. Then the sink reinforces the selected path to the sensor node. This protocol is efficient only with on demand query.

A hierarchical routing protocol improves scalability and minimizes traffic by building clusters. Each sensor transmits its data to its cluster head that is in charge of aggregating them and sending them to the sink. On the contrary, in a flat protocol, all sensor nodes can forward data to the sink if they are in communication range. The most famous hierarchical routing protocol is LEACH [25], where the cluster head is chosen according to its signal strength. Energy consumption is balanced by a random change of cluster heads. Initially, LEACH supports only single hop routing between both a sensor node and its cluster head and between a cluster head and the sink. TEEN [17], is another example of hierarchical protocol. TEEN is also data centric. It builds clusters of different levels until reaching the sink node. The cluster head broadcasts two thresholds to the nodes: only a value of the sensed attribute higher than the hard threshold can force the sensor node to transmit, and if this sensor was already transmitting only changes higher than the soft threshold should be transmitted. This mechanism considerably reduces the number of transmissions. However, TEEN is not adapted to periodic queries. An extension called APTEEN [18] has been proposed.

A geographical routing protocol improves routing by using location information. Each sensor node is assumed to know its location (e.g. with GPS for instance). Hence, the broadcast of a query can be limited to a given region, as in GEAR [19]. In this protocol, a node forwards a packet to its neighbor that is the closest one to the destination. If all neighbors are at the same distance, some neighbor is randomly selected. GAF [20] is another example of geographical routing protocol energy aware. Indeed, each node uses the knowledge of its location provided by its GPS to determine its region in the virtual grid. Nodes located in the same region of the grid are considered equivalent from the routing point of view. Moreover, GAF allows nodes to sleep as long as the existence of an active node per region in the virtual grid is achieved.

Energy criteria taken into account for route selection enables to distinguish between protocols selecting (i) the minimum energy path like [30], (ii) the path avoiding nodes with low residual energy like REAR [32] and (iii) the path of minimum energy while avoiding nodes with low residual energy, like EOLSR [**?**]. This protocol will be presented in Chapter 3.

Multipath routing protocols maintain several paths for the same couple (source, destination). As energy is taken into account, the path minimizing the energy consumed by an end-to-end transmission is usually selected as the primary path. Another path, called the secondary one, is used less frequently, with a probability inversely proportional to its energy cost. The source selects for each data packet to transmit, the path to use according to this probability. Source routing is generally used to force the data packet to follow the path selected by the source. Protocols described in [27] to [39] are multipath routing protocols. On the contrary, with a hop-by-hop routing protocol, a data packet does not contain its path and is routed to the next hop, at each visited node.

A routing protocol can support sleeping nodes. As the sleep state is the least consuming state, making nodes sleep during their inactivity period spares energy. Such protocols have to maintain both network connectivity and application functionalities. As examples, we can cite GAF [20], SPAN [21] and PEN [22]. In SPAN, nodes build a connected backbone such that only nodes belonging to this backbone are active. These nodes are called coordinators. A node decides to become a coordinator using a random backoff delay, function of its residual energy and the number of neighbors it can bridge.

Other classifications exist, like for instance this given in [23] that does not take into account the support of sleeping nodes.

An energy-aware routing protocol can exhibit several criteria. For instance, EAP [24] is hierarchical and allows nodes to sleep, while ensuring the coverage of the monitored region as well as network connectivity.

In the next section, we present the second energy efficient strategy: node activity scheduling.

2.5.2 Node activity scheduling

As the energy consumed in sleeping state is smaller than the energy consumed in any other state as shown in Table 2.1, turning off the sensor radio when it does not receive or transmit data. Hence, keeping sensor nodes in the sleep state, is a good way to save energy. However, this must be accompanied by node activity scheduling to prevent network partition and message loss when some nodes are in sleep state. We can distinguish solutions that:

- are independent of the medium access, and build active node sets.
- depend on the medium access: CSMA/CA, TDMA or hybrid.

2.5.2.1 Solutions independent of the medium access

In this type of solutions, the number of deployed nodes is supposed higher than the optimal number (i.e nodes are redundant). The goal of these protocols is to build a set of active nodes, such that only nodes belonging to this set must be active, all other nodes can sleep to keep their energy while network connectivity and application functionalities are still ensured.

In [7], the authors propose to divide the network nodes in disjoint sets, such that each node set satisfies the network function. These sets are activated successively, and at any time only the nodes of one set are active. All others nodes are in the sleep state. The problem consists in maximizing the number of disjoint sets. It has been shown NP-complete. The protocol proposed may increase network lifetime proportionally to the number of disjoint sets. However, this centralized protocol, assumes that all nodes are synchronized and does not take into account the real node energy consumption: it supposes that the energy consumption is uniform for all nodes belonging to the same set.

Authors in [9] propose to maximize network lifetime in dividing network nodes into non disjoint sets, unlike in [7]. Every set satisfies the network function. They prove that the node participation in several sets may improve network lifetime.

In [8], a distributed and localized solution is proposed. It consists in selecting a connected dominating set of sensor nodes (i.e. each node is either in this subset or is a neighbor of a node in this subset). Only the nodes of this set are active. All other nodes can change their state to sleep mode. A priority is assigned to each node, this priority depends on its residual energy. A node whose function is assured by the dominating set can sleep if and only if:

- the dominating set is connected,
- all its neighbors have at least one neighbor in this set,
- all nodes in the dominating set have a higher priority.

This protocol is localized: each node needs to know information about its one and two-hop neighbors only.

2.5.2.2 Solutions depending on the medium access

With these solutions, any node is allowed to sleep, whenever it is neither transmitting nor receiving. These solutions can be classified in three classes on depend of the medium access:

- **CSMA/CA**. The RTS/CTS exchange preceding any unicast data transmission is used to enable the neighbors of the sender and the receiver to sleep during the communication. In [10], S-MAC, an energy efficient MAC protocol for sensor networks is introduced. The main goal of S-MAC protocol is to reduce energy consumption by using a periodic sleep-wake up cycle, while supporting good scalability and collision avoidance. It consists of three major components: 1) periodic listen and sleep, 2) collision and overhearing avoidance, and 3) message passing. It is based on a CSMA/CA channel access and the RTS/CTS mechanism. Any two nodes exchanging RTS/CTS packets in the listen period need to keep in an active state and start an actual data transmission. Otherwise, all other nodes can enter the sleep mode in order to avoid any wasteful idle listening and overhearing problems. Many other variations of S-MAC are proposed like T-MAC [11] with an adaptive length of active state, D-MAC [12] to reduce the network latency, O-MAC [13] to improve the throughput, etc. However, the RTS/CTS mechanism is not adequate in case of short message which is usually case in wireless sensor networks. This increases the overhead and reduces the protocol efficiency.

- **TDMA**. In TDMA, node transmissions are scheduled and no energy is wasted in collision. In [61], a deterministic solution based on slot assignment named TRAMA is proposed. It consists in three modules: 1) a neighborhood discovery protocol, 2) a schedule exchange protocol and 3) an adaptive election algorithm that selects the transmitter and receiver(s) for each time slot. Only nodes having data to send contend for a slot; notice however, that a node does not know which of its 1-hop and 2-hop neighbors have data to send. The node with the highest priority in its twohop neighborhood wins the right to transmit in the slot considered. Each node declares in advance its next schedule containing the list of its slots and for each slot its receiver(s). The adaptivity of TRAMA to the traffic rate comes at a price: its complexity. To reduce the complexity of TRAMA, FLAMA [62] is introduced. However, this protocol is designed only for data gathering applications in sensor networks based on tree structure. The protocol is simplified both in terms of message exchange and processing complexity. The number of slots allocated by FLAMA to a node with a given traffic rate highly depends on node priority computation.

- **Hybrid**. Z-MAC [97] operates like CSMA under low contention and like TDMA under heavy contention, reducing collisions among two-hop neighbors by means of an initial slot assignment made by DRAND [80]. The goal of Z-MAC is to optimize the bandwidth efficiency of the MAC protocol, selecting CSMA/CA and TDMA when they exhibit the best performance. We can notice that Z-MAC does not allow an immediate acknowledgement of unicast messages. However this acknowledgement is important in wireless communication to confirm the correct reception of the packet. Moreover, since a slot that is unused by its owner can be used by one of its neighbors, the nodes must stay awake in order to be able to receive this message if they are the destination. From the energy point of view, Z-MAC reduces the activity period in the polling cycle enforced by the application but does not allow nodes to sleep during the activity period, what does SERENA (see Chapter 4) whose goal is to maximize network lifetime by scheduling node activity. DRAND [80] assigns slots to nodes in such a way that one-hop and two-hop neighbors have different slots. This randomized algorithm has the advantage of not depending on the number of network nodes but at the cost of an asymptotic convergence.

In the following, wa present the third energy efficient strategy which deals with topology control.

2.5.3 Topology control by tuning node transmission power

Topology control algorithms have been proposed to reduce energy consumption and improve network capacity, while maintaining network connectivity. The key idea of topology control is that instead of transmitting using the maximal power, each node adjusts its power transmission. Thus, energy dissipated in transmission is reduced and a new network topology is created. Topology control algorithms can have several goals:

- Reduce energy consumption (power grows at least quadratically with distance),
- Reduce interference,
- Improve spacial reuse and mitigate the MAC-level medium contention.

The solutions are generally based on one of the three following algorithms minimizing the number of edges in the network graph while ensuring connectivity:

- RNG, Relative Neighborhood Graph, [40], removes any edge directly connecting two nodes if there exists a path of two hops or more between them;

- MST, Minimum Spanning Tree, [41], transforms the network graph in a minimum energy broadcast tree rooted at a given source node giving the shortest path to any other node; only the energy dissipated in transmission is taken into account.
- LMST, Local Minimum Spanning Tree, [42], is a localized algorithm where each node computes its own MST in its neighborhood and only retains those one-hop neighbors on the tree as its neighbors in the final topology. The network topology under LMST is all nodes with their individually perceived neighbors relations.

In [2], the idea is to reduce the transmission range d of every node to minimize the energy dissipated in transmission while keeping a connected topology. It is shown that there exists an optimal transmission range which minimizes energy consumption. But this range depends on the communication type (broadcast or unicast). This solution is localized and distributed, but it does not take into account energy dissipated by reception.

In [3], the TCH (Topology Control with Hitch-hiking) problem is addressed. Hitch-hiking takes advantage of a physical layer that allows combining partial messages containing the same information to decode a complete message. The goal of TCH is to obtain a strongly connected topology minimizing the energy dissipated in transmission. The TCH problem is NP-complete. To resolve this problem, a distributed solution DTCH (distributed TCH) based on 2-hop neighbors information is proposed: each node i increases its transmission power to contribute to the coverage of its 2-hop neighbors j using hitch-hiking. Thus, the i's 1-hop neighbors can reduce their transmission power while keeping a strongly connected network. This solution reduces the complexity of TCH to a polynomial complexity.

An Adaptive Transmission Power Control (ATPC) is proposed in [43]. The goal of ATPC is to achieve energy efficiency and guarantee link quality between neighbors. ATPC allows each node to know the optimal transmission power level to use for each neighbor while maintaining a good link quality. This power is adjusted adapting to spatial and temporal factors by collecting the link quality history. Results show that this protocol is more reliable and energy efficient than the one the maximum transmission power level protocol and also the one using the minimum transmission power level that allows them to reach their neighbors.

In the next section, we will the last energy efficient strategy which reduces the amount of information transferred to spare energy.

2.5.4 Reduce the amount of information transferred

The last energy efficient strategy consists in aggregating information (e.g.; in a data gathering application, a node sends to its parent a single message containing the values transmitted by its children), reducing wasteful transmissions (e.g.; transmission of an information that is already known by the receiver or in which it has no interest) or tuning the refreshment period of control messages (e.g.; neighborhood discovery, topology dissemination, data gathering tree structure). These solutions can be classified in three classes:

- Information aggregation,
- Optimized flooding,
- Tuning the information refreshment frequency.

In the following, we present each class by quoting some works focusing in each class.

2.5.4.1 Information aggregation

Information aggregation is frequently used in data gathering applications, where it enables high benefits. It can use clustering or not.

1. **With clustering**:

   LEACH (Low-Energy Adaptive Clustering Hierarchy) [25] organizes nodes into clusters with one acting as cluster head which is used as router to the base station (i.e.; the data collector). Hence, all members of a cluster transmit their data to the cluster head. Then the cluster head aggregates and compresses all the data received and sent it to the base station. This protocol is localized: nodes organize themselves into clusters. The energy consumption of nodes is balanced by means of a randomized rotation of the cluster heads over time. Performance evaluation results show that LEACH reduces communication energy by as much as 8 times compared to direct transmission.
   Always considering data gathering applications, the authors of [26] have established the conditions under which clustering protocols are energy efficient: the cluster should be based on the similarity of data to transmit. They have also determined the optimal radius of a cluster (i.e.; two hops), assuming that each network area of size $r * r$ (where r is the transmission range) contains exactly one node. Their new protocol, called LNCA, is based on these results. The LNCA cluster set up consists of four steps: 1) exchange of data among neighbors to determine node degrees (i.e.; number of nodes having similar data), 2) exchange of node degrees to select the cluster head, 3) cluster head announcement, 4) join the selected cluster head. Performance evaluation shows that LNCA outperforms LEACH.

2. **Without clustering**: In [4], the MLDA (Maximum Lifetime Data Aggregation) problem is to find a data gathering schedule with a maximum lifetime (duration after which a node has exhausted its energy) for the sensor network which allows data aggregation. In each round, an aggregation tree is created, whose root is the data sink. This tree specifies how data will be collected, aggregated and transmitted to the data sink. To resolve this problem, an integer

program is presented with a linear relaxation. This centralized protocol improves network lifetime but keeps a high time complexity.

The key idea in [5] is to differentiate the quality of data collected from different sensor nodes to balance their energy consumption. In this protocol, the sensor application is assumed to tolerate an error bound E on data aggregation. This error bound is partitioned and a portion e_i of the error bound is assigned to node i. For example, if the sum is applied in the aggregation, we must have $\sum_i e_i \leq E$. A node i sends a new message to the base station only if the difference between the new reading value and the old one exceeds its error bound e_i. So this error bound indicates the frequency of data sending to each node. The portion of error bound allocated to each node is adjusted automatically according to the residual energy to balance the energy consumption and avoid the failure of some sensors faster than others.

2.5.4.2 Optimized flooding

Classical flooding in wireless ad hoc networks leads to a high number of redundant transmissions. A lot of works has been done to optimize network flooding by reducing the number of useless transmissions. We can distinguish three types of optimized flooding:

- Those based on a connected dominating set, where only the nodes belonging to the connected dominating set retransmit the broadcast message. The problem of finding a minimum connected dominating set has been proved NP-hard [63]. Distributed heuristics exist such as [47, 48, 49] where a connected dominating set is built initially and then pruned by removing redundant nodes. Others use the spanning tree of a leader node to assign a rank to each node, such as [51, 52]: depending on its rank and the rank of its neighbors, a node is either dominator or dominatee. If required by network connectivity, some dominatee nodes are included in the dominating set.

- Those using multipoint relays (MPRs). This optimized flooding is used in the OLSR routing protocol to disseminate topology information. The principle is simple. Each node selects its multipoint relays: the multipoint relay set of a node is the minimum set of one-hop neighbors covering all its two-hop nodes. A node N forwards a received broadcast message if and only if this message has a non-null time-to-live and has been received for the first time from a node having selected N as multipoint relay. The benefits brought by this optimization are high in dense networks (see [46]), which is the case of sensor networks. A comparison between connected dominating set flooding and MPR flooding can be found in [50].

- Those based on neighbor gossip, where neighbors exchange the identifiers or the descriptors of the most recently received data. Nodes interested in the corresponding data ask them to the node having these data. For instance, in SPIN (Sensor Protocols for Information via Negotiation) [6], data is described by a descriptor named meta-data which is unique and shorter than the actual data. Before transmitting a message, any node advertises the presence of new data by means of their meta-data. Interested nodes request the data that are finally transmitted only to the interested nodes. Results show that SPIN consumes about 25% less energy than classic flooding. However, some data can be missing, if the message advertising them is lost.

2.5.4.3 Tuning the information refreshment frequency

Energy can also be spared by decreasing the frequency of information refreshment in the network. As a counterpart, the information maintained in the network is less accurate. An interesting tradeoff has been found with the fish eye principle: a fish sees very clearly only what is close, its view strongly decreases with distance. In the Fish eye routing protocol, [44], the information associated with far nodes is refreshed less frequently than this associated with close nodes. This principle has also been applied to scalable extensions of OLSR, [45].

2.6 Cross layering optimization

We can notice that whatever the category of energy efficient strategy considered, solutions can be generic and applied whatever the application (e.g. TRAMA) but not optimal for a given application, or on the contrary be optimized for a specific application (e.g. FLAMA) but of limited applicability. In between these two types of solutions, cross-layering, [64], can help to design generic solutions able to take into account application specificities to maximize network lifetime.

The specificities of wireless ad hoc and sensor networks include:

- dynamically changing network conditions due to mobility or versatility of the propagation conditions on the physical medium,
- network resources of limited capacity: limited bandwidth, limited amount of energy for battery operated nodes, limited processing capability, limited memory, etc,
- radio interferences, etc.

bring new constraints in designing protocols for wireless sensor networks. These protocols must be energy efficient and dynamically adaptive. Moreover, they must satisfy the quality of service desired by the applications. To meet these requirements a cross-layer protocol design that supports adaptivity and optimization across multiple layers of the protocol stack is needed.

In [53], using a cross layer approach, authors study the tradeoff between network lifetime maximization and rate allocation problem by formulating these two problems as a constrained maximization problem. Using Lagrange dual decomposition, the original problem is vertically decomposed into three subproblems: 1) a rate control problem at the transport layer, 2) a contention resolution problem at the MAC Layer, and 3) a cross-layer energy conservation problem. Two fully distributed algorithms are derived for the first two subproblems. For the third subproblem, first they directly derive a partially distributed algorithm, and then propose a fully distributed approximation algorithm using network utility maximization framework.

Nama et al. [54, 55] characterized the tradeoff between maximizing the application performance and lifetime by considering a cross-layer design problem in a wireless sensor network with orthogonal link transmissions, which jointly maximized the network utility and lifetime. The idea is to compute an optimal set of source rates, network flows, and radio resources at the transport, network, and radio resource layers respectively, while jointly maximizing the network utility and lifetime. They show that the cross-layer optimization problem decomposes both horizontally (across nodes) and vertically (across different layers in the protocol stack) into simpler subproblems allowing a fully distributed solution.

In [56, 57], authors investigate the problem of the lifetime maximization in a wireless sensor network under the constraint of the end-to-end transmission success probability. A cross-layering strategy that considers physical layer (i.e., power control), MAC layer (i.e., retransmission control) and network layer (i.e., routing protocol) jointly is adopted. They first present a near-optimal retry limit allocation algorithm for a given routing path. This allocation determines per-hop success probability for each link along the path in order to minimize the total energy consumption while guaranteeing the reliability constraint. Then, an optimal routing and power control algorithm that maximizes the network lifetime is developed. Simulation results reveal that a trade-off relation exists between the network lifetime maximization and the reliability constraint, and that the network lifetime can be increased significantly by employing the proposed algorithms.

2.7 Analysis of node energy consumption distribution

We focus on the distribution of the node energy consumption in a wireless ad hoc network, in order to highlight the main energy costs and then to propose a strategy for improving the network energy efficiency.

In the following we will analyze the node energy consumption distribution in wireless ad hoc and sensor networks. In other words, we will quantify the energy dissipated in transmitting, receiving, overhearing, idle listening and interference during network lifetime. The network lifetime is defined as the time of the first network partition. In other words, it is the time when a flow can no longer reach its destination.

Simulation parameters. Simulations have been performed for different wireless networks. Different parameters of simulation are listed in the following table 2.3.

The OLSR [34] routing protocol that provides the shortest route is chosen. Messages of the routing protocol are not taken into account. All nodes are in active states. User traffic consists of

	Simulation parameter	Value
Configuration	Number of nodes	50-200
	Density	10
	Bandwidth	2Mbps
	Transmission range	250m
	Interference range	500m
Energy	Initial energy	100Joules
	Transmit	1.3Watt
	Receive	0.9Watt
	Idle	0.74Watt
	Sleep	0.047Watt
Traffic	Number of flows	30
	Throughput	16Kbps
	Packet size	512 bytes

Table 2.3: Parameters used for simulation.

30 flows, with randomly chosen sources and destinations. Each result is the average of 10 simulation runs.

Simulation results. Figure 2.1 illustrates the average on all network nodes of the energy dissipated in the different states defined previously.

Furthermore, it clearly appears that in the simulations conditions, the highest part of energy (about 50% in a network of 100 nodes) is surprisingly dissipated in the idle state. The second highest energy cost is due to interferences (about 30%). The third cost is due to overhearing (about 10%). Finally, the energy dissipated in the Transmit and Receive states are small (about 3%). According to these results, we conclude that there are a great amount of energy wasted as the energy consumed in idle, overhearing and interferences. Nevertheless, the energy consumed in transmit and receive messages which is the useful energy is the smallest one. Hence, there are a great work needed to minimize the energy lost and so improve the both the energy consumed in the transmission and the reception of user data and the network lifetime.

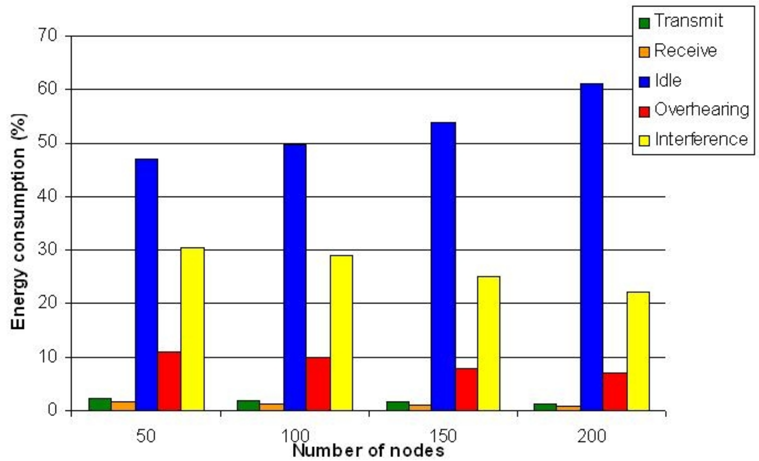

Figure 2.1: Distribution of node energy consumption without sleeping state.

2.8 Discussion

We have presented a brief state of the art related to energy efficient strategies, significantly contributing to maximize network lifetime. We have distinguished between 1) energy efficient routing, 2) node activity scheduling, 3) topology control by adjusting the node transmission power and 4) reducing the amount of information transferred. It is clear that the highest benefits will be obtained by a solution that:

- **combines all energy efficient strategies**. To be energy efficient many solutions integrate more than one strategies. For instance, LEACH which integrates an energy efficient routing protocol and data aggregation, or FLAMA that allows nodes to sleep and aggregates data. It is evident that integrating the four strategies leads to the highest benefit concerning the network lifetime. However, in practice, wireless network constraints can lead to the impossibility to use some one solutions. For example, if nodes cannot tune their transmission power, the topology control cannot be integrated. In addition, if the sleeping mode is not supported by lower layers, node activity scheduling becomes inapplicable in these cases. Hence, the goal is to prolong the network lifetime as much as possible taking into account the network constraints and application requirements.

- **minimizes the complexity**. Wireless sensor networks have many physical limits: limited memory, limited processing capacity, limited bandwidth and energy. Consequently, energy efficient solutions supported in such networks must have a low complexity. Moreover, the overhead induced by any solution in the wireless ad hoc and sensor networks, is an important criterion to decide on a solution. This overhead must be as low as possible.

- **is distributed and localized**. Centralized solutions may have a good performance for a small network. However, in large wireless ad hoc and sensor networks, they induce many disadvantages:

- Vulnerability. All network information is stored in the central node. The failure of this node can lead to the failure of all the network.
- Important overhead. All information must be collected and sent to the central node which is in charge of executing algorithms. Results will then be will be sent to all nodes in the network
- Not scalable. The complexity increase with the number of nodes.

For these reasons, a distributed solutions based on local information is more adapted to wireless ad hoc and sensor networks. Moreover, distributed solutions must be localized. In such solutions, each node executes the solution based on local information.

In all cases, the target of all these solutions is to maximize the network lifetime. The ideal solution is to avoid all useless states from the application point of view (idle sate, interferences and overhearing) that waste energy. Hence limiting the energy consumption to the transmit and receive states. To be adopted such a solution should be decentralized, localized, self-adaptive to traffic and topology changes, while having a low complexity in terms of messages exchange, memory and processing power required.

2.9 Conclusion

In this chapter, we have first provided a summary of energy saving solutions proposed for wireless ad hoc and sensor networks. We have distinguished between 1) energy efficient routing, 2) node activity scheduling, 3) topology control by adjusting the node transmission power, 4) reducing the amount of information transferred. Using cross layering design in wireless ad hoc and sensor networks is a very promising approach to improve network lifetime. In fact, the knowledge of applications specificities as well as lower layer information (SINR, the power of received signal, etc.) can participate to improve the performances (throughput, reliability, etc.) as well as the lifetime of wireless ad hoc and sensor networks. The analysis of the node energy consumption distribution provided at the end of this chapter shows that a large amount of energy is wasted in useless states from the application point of view. Our target is to minimize as much as possible this energy wasted. Consequently, the network lifetime can be improved and the amount of user data delivered can be increased.

Chapter 3

Energy efficient routing

3.1 Introduction

In order to maximize network lifetime, several directions have been explored as exposed in Chapter 2. Among them there is energy efficient routing. In this chapter, we show how to extend the standardized $OLSR$ routing protocol, in order to make it energy efficient.

This chapter is organized as follows. In Section 3.2, we present some works related to energy efficient routing protocols. We then focus more particularly on the $OLSR$ routing protocol. In Section 3.3, we show how to extend this protocol to make it energy efficient. This extension is based on an energy model (see Subsection 3.3.4) and includes three algorithms. The first one is the selection of multipoint relays taking residual energy into account (see Subsection 3.3.5). The second one computes energy efficient routes (see Subsection 3.3.6). The third one deals with the optimized broadcasts (see Subsection 3.3.7). In Section 3.4, we compare EOLSR with a routing protocol minimizing energy and with another one maximizing the number of packets exchaged. We also compare EOLSR with multipath routing protocols with disjoint nodes or with disjoint links. In Section 3.5, we show how $EOLSR$ can be adapted to specific applications, more precisely data gathering applications. Finally, we conclude in Section 3.6 .

3.2 Related work

Energy aware routing protocols establish energy efficient routes to maximize network lifetime. These protocols run in wireless sensor networks subject to radio interferences. In such networks, a very interesting complexity result has been established in [36]: because of radio interferences the selection of a unicast route ensuring that each node has sufficient residual energy is NP-hard. This result is still valid, even if radio interferences were limited to a single hop.

Before proposing an extension of the OLSR routing protocol to make it energy efficient, we will first briefly present works related to energy efficient routing. In Chapter 2, we have made a state of the art and given a classification of different energy-aware routing protocols. In this section, we will focus on different ways to compute efficient routes presented in the literature. We can distinguish two families of energy efficient routing protocols: 1) multipath routing protocols in which each source maintains several routes to the destination and 2) the adaptive hop-by-hop routing protocols in which only the route that minimizes the energy is maintained.

3.2.1 Multipath routing protocols

Multipath routing protocols rely on the following observation: using the lowest energy path leads to energy depletion of nodes along this path and in the worst case may lead to network partition. To avoid this problem and improve network lifetime, these protocols use multipath routing instead of the minimum energy path.
In [27], a multipath routing, energy aware, is proposed to maximize network lifetime. It is a reactive routing protocol triggered by the data sink. It consists in finding all paths between source and destination according to a metric which takes into account 1) the energy consumed by the transmission and reception of the packet and 2) the residual energy of nodes. Paths that have a cost higher than a given threshold are discarded. At each path kept is assigned a probability inversely proportional to the cost of the path. To route a flow packet, one path will be randomly chosen by the source according to its probability. This protocol ensures load balancing by means of multipath routing. Moreover, considering residual energy of nodes to compute paths allows to avoid nodes with low energy capacity.
In [29], two variants of minimum energy multipath routing are considered to improve network lifetime and reliability against nodes and links failures. These two variants are: 1) multipath with disjoint links and 2) multipath with disjoint nodes. The authors show that link disjoint path routing is more energy efficient and node disjoint path routing is more reliable. However, their model does not take interferences into account. Performance evaluations in [29, 39] show that the relative benefit of maintaining one additional path strongly decreases with the number of paths maintained, whereas the complexity increases a lot. That is why, maintaining two paths is generally considered sufficient.

3.2.2 Adaptive hop-by-hop routing protocols

The second category of energy efficient routing protocols corresponds to adaptive hop-by-hop routing protocols. We can distinguish three families of energy efficient routing protocols:

- *the protocols selecting the path consuming the minimum energy.* The advantage is that each transmission of a packet from its source to its destination minimizes the energy consumed. We can cite for example [30] and a more sophisticated protocol [31]. This routing protocol [31] computes the additional energy dissipated by one flow when routed on a given path, taking into account the SINR (Signal over Interference and Noise Ratio) and the energy lost by radio interferences. Then, it uses the Dijkstra algorithm to find the path which minimizes this additional energy. The advantage of this protocol is that it takes into account the impact of the transmission of the flow in the interference area. However, this protocol is complex and requires that all nodes know the global topology of the network. Furthermore, such protocols use always the same nodes (those minimizing the energy consumed) without any consideration on their residual energy. Consequently, these nodes will exhaust their battery more quickly than the others and the network lifetime is not maximized.

- *the protocols selecting the path visiting the nodes with the highest residual energy*, such a REAR (Reliable Energy Aware Routing)s [32]. REAR ensures that each flow is ensured to have enough energy on the selected path: nodes with low residual energy are avoided. REAR is a reactive protocol triggered by the data sink. To achieve its goal, the amount of energy needed by the end-to-end transmission of the flow must be reserved in each intermediate node. Besides, to improve reliability a second path is computed to route a flow in case of first path failure. This path is disjoint from the first path and has enough energy to route the flow. But, there is no energy resource reservation on this second path. This protocol ensures the energy necessary to route a flow in the intermediate nodes but it does not take into account energy dissipated by the reception of packets and the interferences. However, the path selected does not minimize the energy needed to transmit a flow packet from its source to its destination. Hence, the network lifetime may not be maximized.

- *the hybrid protocols selecting the path with the minimum cost, where the cost takes into account the residual energy of each visited node (and possibly its neighbors) and the energy consumption of a packet on this path.* These protocols avoid the problems encountered by the protocols of the two previous categories by weighting the factors used in the cost computation. We can cite for instance [33].

In the next section, we will present an energy efficient routing protocol belonging to the third family of the adaptive hop-by-hop routing protocols family based on $OLSR$.

3.3 EOLSR: Energy efficient extension of OLSR

3.3.1 OLSR functioning overview

The OLSR (Optimized Link State Routing) protocol, RFC 3626 [34], has been optimized for MANETs (Mobile Ad Hoc Networks). It can also be used in wireless sensor networks. OLSR is a proactive routing protocol where nodes periodically exchange topology information in order to establish a route to any destination in the network. It is an optimization of a pure link state routing protocol, based on the concept of multipoint relays (MPRs). First, using multipoint relays reduces the size of the control messages: rather than declaring all its links in the network, a node declares only the set of links with its neighbors that have selected it as "multipoint relay". The use of MPRs also minimizes flooding of control traffic. Indeed only multipoint relays forward control messages. This technique significantly reduces the number of retransmissions of broadcast messages.

OLSR provides two main functionalities:

- *Neighborhood discovery.* Each node detects its one-hop and two-hop neighbors by means of periodic Hello messages. Each node independently selects its own set of MPRs among the one-hop neighbors in such a way that its MPRs cover all its two-hop neighbors.
- *Topology dissemination.* Each node also maintains information about the network topology obtained from Topology Control (TC) messages, periodically generated by MPRs and disseminated by MPR flooding. The routing table is computed using Dijkstra's algorithm. This table provides the shortest route (i.e. the smallest hop number) to any destination in the network.

Performance evaluation results [35] showing that a MANET with OLSR routing achieves very satisfying performances.

3.3.2 Why OLSR is not energy efficient?

OLSR does not take into account energy in its design. In fact,

1. the MPR selection algorithm in OLSR does not take into account energy but only the number of nodes covered at two-hop. However, these MPRs are the intermediate nodes along a route computed by OLSR.
2. the cost used by OLSR to build routes is the number of hops. Consequently, a route computed between a source and a destination is the shortest route. However, the shortest route does not necessarily minimize the energy consumed by the end-to-end packet transmission.
3. using MPRs nodes as intermediate nodes in routes, and the shortest routes could leads to always utilize the same nodes for routing messages. Therefore, these nodes can exhaust their energy very quickly with regard to other nodes in the network and thus leads to the network patitionning.

However, we must indicate that OLSR optimizes the network flooding by means of MPRs. In fact, MPRs are used in *OLSR* to optimize network flooding by means of the following forwarding rule: *a node forwards once a broadcast message with a non-null time-to-live only if it has received this message for the first time from a node that has selected it as MPR.*
Nevertheless, the fact that OLSR does not, in its design, take into account energy motivates us to propose EOLSR: Energy efficient OLSR.

3.3.3 Principles of EOLSR

EOLSR, is the energy efficient extension of OLSR. Like OLSR, it consists of two main functionalities Neighborhood discovery and Topology dissemination. Our goal is to make the OLSR routing protocol energy efficient in order to maximize the network lifetime by:

- Minimizing the energy consumed by a packet transmission from its source to its destination. The transmission and reception is a source of energy consumption, and optimizing its number could optimize the energy consumption.
- Balancing load between nodes of the network. Using the same nodes to route messages exhausts the batteries of these nodes. As a consequence, they will fail more quickly than others. This could lead to network partitionning or some application functionalities are no more assured (i.e. zone no more monitored).
- Avoiding nodes with a low residual energy.
- Reducing the overhead.

To achieve these goals, the design of EOLSR requires to address the following topics:

1. The choice of the energy model. The energy model chosen for EOLSR is very important for the design of the protocol. In fact, the routing algorithm will be based on this energy model to build energy efficient routes.
2. Energy aware MPRs selection: as the intermediate nodes in a path are MPRs, its selection must also take energy into account.
3. Computation of energy aware routes: energy must be a criterion of route selection and the end-to-end energy consumption must be minimized. The route computation is based on the energy model chosen.
4. Broadcast optimization. message flooding is very consuming for network resources and even energy. Thus it must be minimized and optimized.

In the next section, we will propose and discuss solutions for each of these topics. The best solution will be chosen to design EOLSR.

3.3.4 Energy consumption model

Energy efficient routing is aware of energy dissipated during a transmission. In all cases, an energy consumption model is needed to conclude in favor of an increase in network lifetime. Indeed, this model can be used by the routing protocol itself to take decisions based on energy, and/or by the protocol designers to evaluate the performances of their protocol.

As reported in Chapter 2, Section 2.4.1, a wireless node's radio can be in one of the following four states: $Transmit$, *Receive*, *Idle* or *Sleep* and each of which consumes different levels of energy. In our work, we suppose that all nodes transmit with the the same transmission power allowing to reach nodes at the maximum transmission range. Moreover, as we are interested in the additional energy spent during the transmission of a flow packet from its source to its destination with reference to the idle state, the values used for transmission power and reception power are calculated as follows:

$$P_{trans} = P_{transmit} - P_{idle}. \tag{3.1}$$

$$P_{recv} = P_{receive} - P_{idle}. \tag{3.2}$$

In Table 3.1, we report the reference values of $P_{transmit}$, $P_{receive}$, and P_{idle} taken from a Lucent silver wavelan PC card.

State	Power value	Power Increment
Transmit	$P_{transmit} = 1.3\text{W}$	$P_{trans} = 0.56\text{W}$
Receive	$P_{receive} = 0.9\text{W}$	$P_{recv} = 0.16\text{W}$
Idle	$P_{idle} = 0.74\text{W}$	
Sleep	$P_{sleep} = 0.047\text{W}$	

Table 3.1: Power value in each radio state

Now, we can determine the energy dissipated in transmitting (E_{trans}) or receiving (E_{recv}) one packet (this energy is the additional energy consumed when transmitting or receiving packet with reference to the idle state). Let *Duration* denote the transmission duration of a packet. We have:

$$E_{trans} = P_{trans} * Duration. \tag{3.3}$$

$$E_{recv} = P_{recv} * Duration. \tag{3.4}$$

When a transmitter transmits a packet to next hop, because of the shared nature of wireless medium, all its neighbors receive this packet even it is intended to only one of them, we speak here about overhearing. Moreover, each node located between transmitter range and interference range receives this packet but cannot decode it. Hence, we can refine the Receive state in three substates:

- Receive: when this node is the packet destination,
- Overhear: when this node is an one-hop neighbor of the sender and is not the destination,
- Interference: when this node is a two-hop neighbor of the sender.

Interference and overhearing generate loss of energy. Some protocols have taken into account both problems, others have considered only one of them. So energy dissipated by a transmission by sender i is calculated as follows [37]:

$$cost_{transmission}(i) = E_{trans} + n * E_{recv}, \tag{3.5}$$

where n represents:

- the number of one-hop non-sleeping neighbors of transmitter i, if the protocol takes into account only overhearing,
- or all non-sleeping nodes belonging to the interference zone of the transmitter i, if the protocol takes into account energy dissipated by overhearing and interferences.

This cost indicates the quantity of energy consumed by a packet of the flow to reach the next hop.

The energy dissipated by an end-to-end transmission over a path P can be computed as follows:

$$cost(flow) = \sum_{i \in sender(flow)} cost_{transmission}(i), \tag{3.6}$$

where i is a sender of $flow$ on its path P.
In the next section, we will present different manner or strategies to take into account energy on the selection of the MPRs.

3.3.5 Energy efficient selection of MPRs

Energy efficient MPRs selection allows each node N in the network to select a subset of its one hop neighbors, according to their residual energy, that permits to reach any two hops neighbors through a path with enough energy. The energy efficient MPR is named EMPR. We keep MPR to denote an MPR selected by the $OLSR$ protocol.
To enable a selection of multipoint relays energy-aware, additional information about energy has to be included in the $Hello$ message (see Section 3.3.8.1). We adopt the usual assumption that interferences are limited to two hops from the transmitter.

3.3.5.1 Presentation of EMPR selection strategies

We now focus on the selection of EMPRs. Let N be the node selecting its EMPRs. Let M be a 1-hop neighbor of N covering at least one two-hop neighbor of N, hence M is an EMPR candidate of N. Let $E_R(M)$ denotes the residual energy of node M. We propose and study three variants depending on the nearby nodes considered for the computation of the minimum residual energy.

1. ***E* strategy**. This strategy considers only $E_R(M)$, the residual energy of the EMPR candidate, M;

2. ***M1E* strategy**. This strategy considers the weighted residual energy of the EMPR candidate M and its 1-hop neighbors:

$$min(\frac{E_R(M)}{P_{trans} + P_{rcv}}, min_{D \in 1hop(M)}(\frac{E_R(D)}{2 * P_{rcv}})). \tag{3.7}$$

   The weights of $E_R(M)$ and $E_R(D)$ take into account the role played by the nodes M (one-hop neighbor of N) and D (two-hop neighbor of N) in a transmission from N, the node performing the EMPR selection, to D, one of its two-hop neighbors, via the node M. It represents the maximum transmission duration that can be sustained;

3. ***M2E* strategy**. This strategy considers the weighted residual energy of the EMPR candidate M and its 1-hop and 2-hop neighbors:

$$min(\frac{E_R(M)}{P_{trans}+P_{rcv}}, min_{D\in 1hop(M)}(\frac{E_R(D)}{2*P_{rcv}}), min_{D\in 2hop(M)}(\frac{E_R(D)}{P_{rcv}})). \tag{3.8}$$

3.3.5.2 Algorithm of EMPR selection

The EMPR selection algorithm, performed by any node N, is the following:

Algorithm 1 Algorithm of EMPR selection

N inserts in the set $Uncovered$ all its 2-hop neighbors

N orders its one-hop neighbors M by decreasing order of the selection criterion:

- In case of the strategy E: the residual energy of M, (i.e.; decreasing order of $E_R(M)$),
- In case of the strategy $M1E$: the minimum 1-hop weighted energy computed according to the equation 3.7 ,
- In case of the strategy $M2E$: minimum 2-hop weighted energy computed according to the equation 3.8,

Let $\mathcal{N}_1$ be this ordered set.

repeat

    Let M the first node in $\mathcal{N}_1$

    if M allows to reach at least one node in $Uncovered$ **then**

        N selects M as EMPR.

        N removes from $Uncovered$ all nodes covered by M.

    end if

    N removes M from $\mathcal{N}_1$

until $Uncovered$ is empty

We can notice that the algorithm of EMPR selection tends to share the energy consumption between the different nodes. This selection avoids that nodes deplete their battery more quickly than others.

However, with regard to the native MPR selection algorithm, this new algorithm introduces more frequent route changes. In fact, the residual energy of nodes decreases continuously which induces frequent changes in the EMPR selection. To limit these changes, we use two thresholds

ThresholdEMPREnergy and **ThresholdMinEnergy**. According to these two thresholds, we change the EMPR selection only if the two following conditions are met:

1. there is at least one 2-hop neighbor D such that :

$$\frac{E_R(new\_EMPR) - E_R(old\_EMPR)}{E_R(old\_EMPR)} > ThresholdEMPREnergy, \tag{3.9}$$

   where $E_R(new\_EMPR)$ represents the residual energy of the new node selected as EMPR to cover D, and $E_R(old\_EMPR)$ represents the residual energy of the previous node selected as EMPR to cover D. For example, in simulation, we take $ThresholdEMPREnergy = 10\%$.

2. and the residual energy of the new EMPR is sufficient: $E_R(new\_EMPR) \geq ThresholdMinEnergy$. This avoids frequent changes when the residual energy of an EMPR tends to 0.

Each node computes its EMPRs before sending its periodic $Hello$. If these two conditions are met for at least one 2-hop neighbor, the previous set of EMPRs is changed by the new set.

3.3.5.3 Performance evaluation of EMPR slection

In this section, we compare the average number of EMPR selected per node according to the three strategies with the average number of MPR per node.

Remark 1. When a QoS (Quality of Service) criterion, such as energy, local available bandwidth or medium access delay, is used to select QoS MPRs (i.e.; EMPRs when the energy criterion is used), the following result has been proved in [38]: *the average number of neighbors selected as QoS MPRs is in $O(n^{1/3}log(n))$, where n is the average number of neighbor per node, whereas the average number of MPRs selected according to the native procedure in OLSR is in $O(n^{1/3})$.* The average number of QoS MPRs selected per node is increased by a $log(n)$ factor.

Simulation parameters:
In the following simulations, nodes are uniformly distributed in the network area. Table 3.2 summarizes simulation parameters used in these series of simulation. The powers used are those given in Table 3.1.

The transmission range is equal to 250m. Interferences are limited to 500m. Results reported in this section are the average of 10 simulations.

First, we evaluate the average number of EMPRs per node as a function of network density, with the three selection variants. The number of nodes is set to 100. Similarly, the network density (i.e.; the average number of one-hop neighbors per node) being fixed to 10, we study the impact of the node number on the average number of EMPRs per node. In both cases, the native MPR selection is used as a reference. Simulation results are illustrated in Figures 3.1 and 3.2.

	Simulation parameter	Value
Configuration	Number of nodes	50-200
	Density	10-30
	Bandwidth	2Mbps
	Transmission range	250m
	Interference range	500m
Energy	Initial energy	uniformed distributed in [20, 60]Joules
	Transmit	1.3Watt
	Receive	0.9Watt
	Idle	0.74Watt
	Sleep	0.047Watt

Table 3.2: Parameters used for simulation.

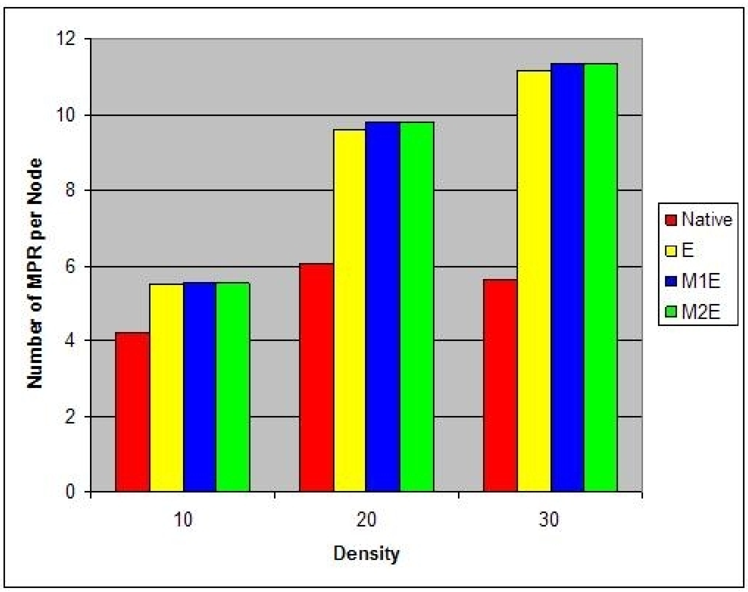

Figure 3.1: Number of multipoint relays per node, function of network density.

Simulation results confirm Remark 1: the number of EMPR per node is higher than the number of MPR per node for the three proposed strategies. We can also notice that the difference increases with network density and does not depend on node number. Furthermore, with the power values taken, the $M1E$ and $M2E$ selections give the same results. The reason is that the criterion used to sort the nodes leads to the same value with both $M1E$ and $M2E$. Indeed the third term in formula 3.8 (see Section 3.3.5) is negligible compared to the first two terms. So the use of $M1E$ is more interesting than $M2E$ because it is less complicated to compute and needs less information from the network.

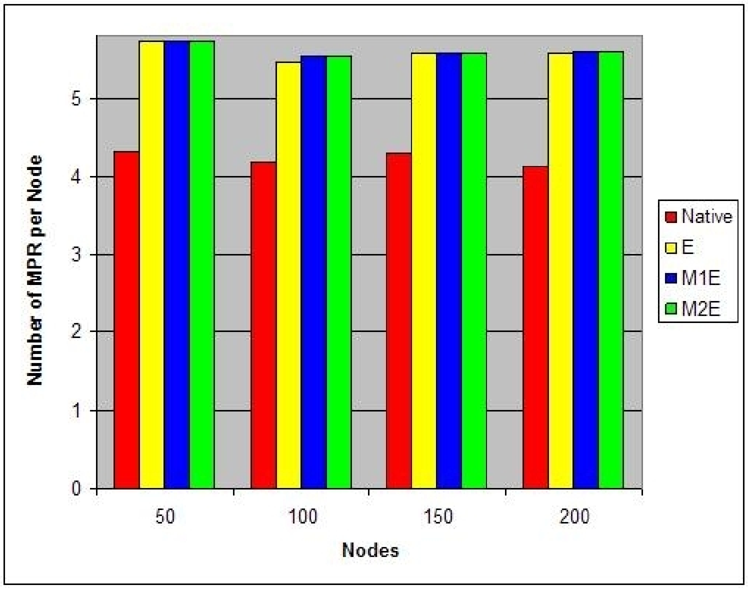

Figure 3.2: Number of multipoint relays per node, function of node number.

Comparing our three proposed strategies, E, $M1E$ and $M2E$, we notice that E has the smallest number of nodes selected as EMPRs. However, in this stage, it is too early to decide the EMPR selection strategy that we will adopted for EOLSR. More performance evaluation is given in Section 3.3.8 which highlights the benefit brought by these strategies on network lifetime.

3.3.6 Routing algorithm for $EOLSR$

Our idea to maximize network lifetime, is to minimize the energy consumed by the end-to-end transmission of a flow in selecting the path, build from EMPRs, consuming the minimum energy. For that, we will use the cost given in the equation 3.6:

$$cost(flow) = \sum_{i \in sender(flow)} cost_{transmission}(i), \tag{3.6}$$

to compute routes for flows. Our target is to keep the complexity of Dijkstra (the same complexity of OLSR). However, instead of using the number of hop between source and destination to select the shortest route (i.e.; every link has a cost of one), as done in $OLSR$, we will use $cost(flow)$ as the criterion to choose the best path where:
every link $i \rightarrow j$ has a cost equal to $cost_{transmission}(i)(equation\ 3.5)$

$$cost_{transmission}(i) = E_{trans} + n * E_{recv}. \tag{3.5}$$

3.3.6.1 Computation of energy efficient routes

As previously said, $EOLSR$ selects the route of minimum energy cost. This cost depends on the network type. $EOLSR$ distinguishes the networks where:

- Nodes do not sleep: we distinguish two cases:

  1. Nodes are in the idle state when the medium is idle, like in IEEE 802.11. In this case, the energy dissipated in receiving + overhearing + interferences takes into account the number of nodes in the 1-hop and 2-hop neighborhood of each sender of the considered flow.

$$cost(flow) = \sum_{i \in sender(flow)} (E_{transmit}(i) + \sum_{j \in \mathcal{N}_1(i) \bigcup \mathcal{N}_2(i)} E_{receive}(j)) \qquad (3.10)$$

  2. Nodes stay in the receive state when the medium is idle, like in $ZigBee$. In order to select the route maximizing network capacity, we take the smallest route.

$$cost(flow) = \sum_{i \in sender(flow)} E_{transmit}(i) \qquad (3.11)$$

  This is equivalent to compute the shortest route with the smallest number of hops.

- Nodes sleep as with $SERENA$, node activity scheduling algorithm that will be described in Chapter 4. A node is allowed to sleep when neither itself nor its 1-hop neighbors have a message to transmit. Hence, only 1-hop neighbors have to be taken into account in the energy cost, leading to:

$$cost(flow) = \sum_{i \in sender(flow)} (E_{transmit}(i) + \sum_{j \in \mathcal{N}_1(i)} E_{receive}(j)) \qquad (3.12)$$

If several routes dissipate the same energy, the shortest one is chosen.

3.3.7 Optimized broadcasts

In wireless networks, we usually need to broadcast messages to maintain network functionalities or protocols performances. However, flooding messages are very greedy on network resources. Consequently, optimizing the number of broadcast messages is very important and influences network performances.
As we can notice in simulation results given in Section 3.3.5.3, the number of EMPRs is greater than the number of MPRs. The idea is to use the MPRs nodes to forward broadcast messages and the EMPRs as intermediated nodes on routes. Hence, we keep the optimization of network broadcasts used in $OLSR$ to optimize network flooding by means of the following forwarding rule:
a node forwards once a broadcast message with a non-null time-to-live only if it has received this message for the first time from a node that has selected it as MPR.

We evaluate the gain obtained by using the optimized broadcast in $EOLSR$. We compare the network lifetime and the number of TC received and forwarded in the network when:

- either MPRs forward TCs messages. Thus EMPRs generate and send TCs messages and the MPRs forward these messages.
- or the EMPRs forward TCs messages. Thus EMPRs generate, send and forward TCs messages.

Simulation parameters. The network simulator ns2.31 [109] is used. Table 3.3 summarizes simulation parameters used in this series of simulation. User traffic consists of 30 flows, with randomly chosen sources and destinations. Each result is the average of 10 simulation runs. The network lifetime is defined as the first node failed because lack of energy.

	Simulation parameter	Value
Configuration	Number of nodes	50-200
	Density	10
	Bandwidth	11Mbps
	Transmission range	250m
	Interference range	500m
Energy	Initial energy	50Joules
	P_{trans}	0.56Watt
	P_{recv}	0.16Watt
Traffic	Number of flows	30
	Throughput	16Kbps
	Packet size	512 bytes
EOLSR parameter	$Hello$ period	2 seconds
	TC period	5 seconds

Table 3.3: Parameters used for simulation.

Figure 3.3 and Figure 3.4 represent the number of TCs received per node and the number of TCs forwarded respectively in TC period.

As expected, using the MPRs for forwarding TCs messages decreases considerably both the number of TCs received per node and the number of TCs forwarded. Considering a network of 150 nodes, this increase reaches 22% in number of TCs received per node as depicted in Figure 3.3. In Figure 3.4, the increase reaches 27% for the number of TCs forwarded in the network. Both results

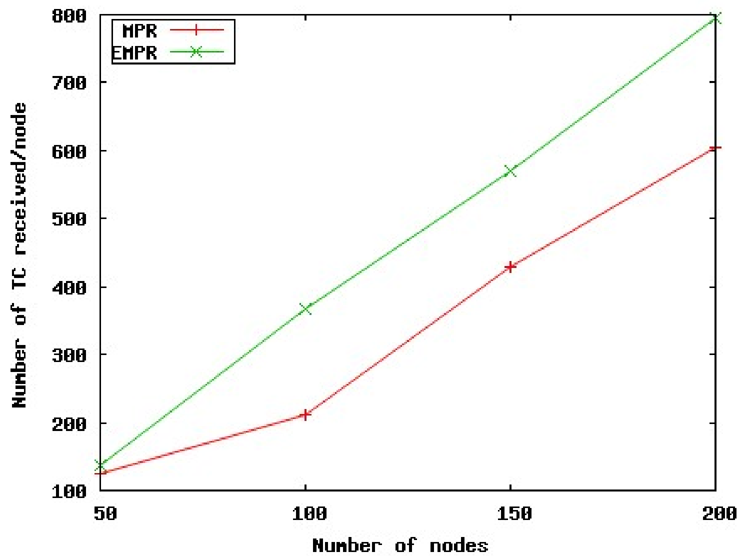

Figure 3.3: Number of TCs received per node

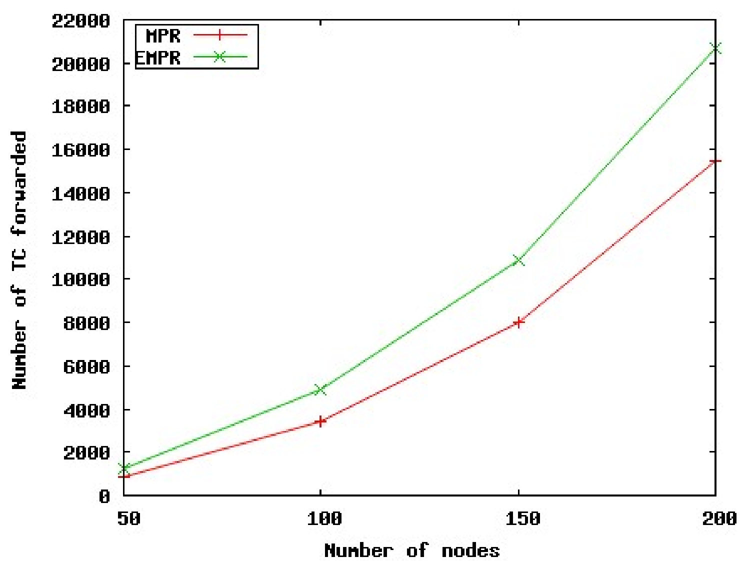

Figure 3.4: Number of TCs forwarded in the network

are explained by the fact that the number of MPRs is smaller than the number of EMPRs.
However, our first target is to maximize network lifetime. Consequently, we consider the impact of the optimized broadcast on the energy efficiency of $EOLSR$ to decide wether to accept or reject this approach.
Figure 3.5 represents the network lifetime with and without optimized broadcast. It shows that using the $MPRs$ to forward TCs messages extends the network lifetime by 10%.

Finally, using the $MPRs$ for optimizing broadcasts, on the one hand, reduces the overhead induced by $EOLSR$ by decreasing the number of TCs messages in transit in the network. On

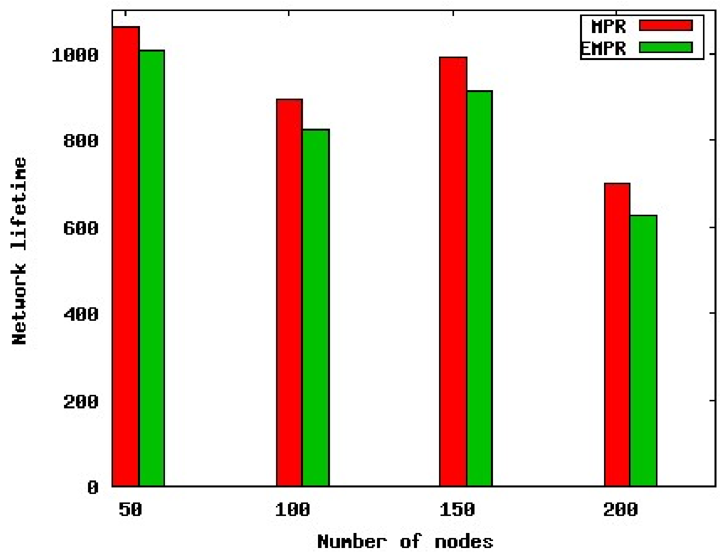

Figure 3.5: Network lifetime with and without the optimized broadcast

the other hand, it improves the performance of $EOLSR$ by increasing network lifetime. Moreover, maintaining and selecting the $MPRs$ and the $EMPRs$ do not increase the complexity of the routing protocol. In fact, each node computes these two sets ($MPRs$ set and $EMPRs$ set) based on local stored neighborhood information. No field is added to $Hello$ or TC messages to achieve that.

3.3.8 EOLSR design

As said et the beginning of this section, EOLSR consists of:

1. **Energy consumption model**. Since information concerning energy consumption is generally localized in lower layers (MAC or Physical layer), we must adopt an energy consumption model in the network layer in order to evaluate the energy consumption and to propose an energy efficient routing protocol. This model is presented in Section 3.3.4.

2. **Energy efficient MPR selection: EMPR**. In a second step, we have focused on EMPR selection. We have proposed three strategies. We have given in Section 3.3.5.2 a first series of performance evaluation that compare the number of EMPR selected according to the three proposed strategies. It shows that the number of EMPR selected according to the E strategy provides the smallest number of EMPRs with regard to $M1E$ and $M2E$ strategies. In the following, we give a second series of performance evaluation that compare the network lifetime provided by the three EMPR selection strategies to decide which strategy we use for EOLSR: **Impact of EMPR selection strategies on network lifetime.** For these simulations, we use the simulation parameters presented in Table 3.3.

   We now compare the three EMPR selection strategies presented in 3.3.5 to choose the one maximizing the network lifetime. Results are shown on Figure 3.6.

First, we observe that $M2E$ and $M1E$ EMPR selection strategies provide the same result. We notice that $EOLSR$ obtains the best performance when used with $M1E$ or $M2E$. That is because, unlike E, $M1E$ takes into account the energy dissipated in packet transmission and reception on the 1-hop neighbors of the transmitter.

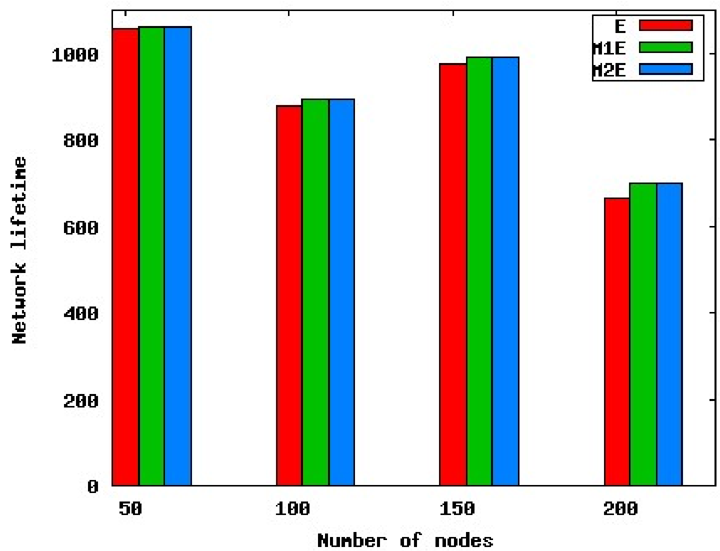

Figure 3.6: Impact of the EMPR selection strategy on the network lifetime

As in our case, both $M1E$ and $M2E$ provide the same results and the criterion used in $M1E$ is simpler to compute than in $M2E$ and requires less additional information in the $Hello$ messages, we use $M1E$ as EMPR selection strategy for EOLSR.

3. **Computation of energy aware routing**. The routes in EOLSR are built with EMPR as intermediate nodes. to minimize the end-to-end energy consumption, the cost used to compute these routes is the energy cost computed as presented in Section 3.3.6.

4. **Optimized broadcasts**. To reduce the number of broadcast messages of the routing protocol (TC message) as well as broadcast user messages, we propose to use the forwarding rule used in classical EOLSR using only nodes selected as MPR.

3.3.8.1 Control messages in $EOLSR$

We notice that EOLSR, the energy efficient extension of $OLSR$ does not need additional messages. Only additional energy information is included in both $Hello$ and TC messages.
In order to select the EMPRs, the $Hello$ messages include the following additional information: the residual energy of the sending node, and the minimum residual energy of the one-hop nodes.
In order to compute the energy cost of a flow, we need to know the number of nodes up to two-hop of the node considered (see Eq. (3.5)). Hence, the TC messages include the number of nodes belonging to the interference area of the TC originator.

3.4 Performance evaluation of $EOLSR$

In this section, we evaluate the performance of $EOLSR$. We suppose that all nodes are in active state. Hence, the equation 3.10 ($cost(flow) = \sum_{i \in sender(flow)} (E_{transmit}(i) + \sum_{j \in \mathcal{N}_1(i) \bigcup \mathcal{N}_2(i)} E_{receive}(j)))$ is used to build routes. Then, we compare the performance of $EOLSR$ in terms of network lifetime and data delivered with:

- two possible variants of energy efficient routing based on OLSR: the first variant selects the path consuming the minimum energy and the second variant selects the path visiting the nodes with the highest residual energy.

- two multipath routing protocols: the first computes two paths with different nodes, and the second computes two paths with different links.

3.4.1 Comparison of $EOLSR$ with $MinEnergy$ and $MaxPacket$

In this section, we consider $MinEnergy$ and $MaxPacket$ two energy efficient routing variants based on $OLSR$ and compare their performances with $EOLSR$.

3.4.1.1 Presentation of $MinEnergy$ and $MaxPacket$

The two energy efficient routing variants considered are:

- $MinEnergy$. The idea of $MinEnergy$ is to select the route that minimizes the energy consumed by the end-to-end packet transmission without considering nodes residual energy. It consists in $EMPR$ selection algorithm and an energy efficient route computation.

  - For the route computation, we use, as $EOLSR$, the equation 3.10:

    $$cost(flow) = \sum_{i \in sender(flow)} (E_{transmit}(i) + \sum_{j \in \mathcal{N}_1(i) \bigcup \mathcal{N}_2(i)} E_{receive}(j)).$$

    The Dijkstra algorithm is used to compute the route minimizing the energy cost given by the equation 3.10 presented above.

  - Concerning the $EMPR$ selection, the routes computed by this algorithm to reach the two-hop neighbors go through the EMPR nodes. In other words, the EMPR nodes of any node N are the one-hop neighbors that minimize energy consumed by a transmission from a considered node N to its two-hop neighbors.

- $MaxPacket$. The idea of $MaxPacket$ is to select routes visiting nodes with the maximum residual energy without considering the energy cost of this route.

  - It keeps the same $EMPR$ selection strategies as $EOLSR$ (based on $M1E$).

– For the route building, $MaxPacket$ computes routes as follows:

$$cost(path) = min_{i(sender(flow)}(min_{M\in 1hop(i)}(\frac{E_R(M)}{P_{trans}+P_{rcv}}, min_{D\in 1hop(M)}(\frac{E_R(D)}{2*P_{rcv}}))).$$

The target is to maximize $cost(path)$. That means that the selected route between one source and one destination maximizes the number of packets that can be transmitted between this source and this destination.

3.4.1.2 Comparative performance evaluation

In this section, we give a comparison of $MinEnergy$, $MaxPacket$ and $EOLSR$ in terms of network lifetime and the quantity of data delivered to the destination. In this series of simulation, we use the ns2 simulator and the simulation parameters given in Table 3.3. The network lifetime is defined by the time of first node failure.
Figure 3.7 presents the network lifetime when varying the number of nodes in the network from 50 to 200. We notice that $MaxPacket$ presents the smallest network lifetime. This is because it

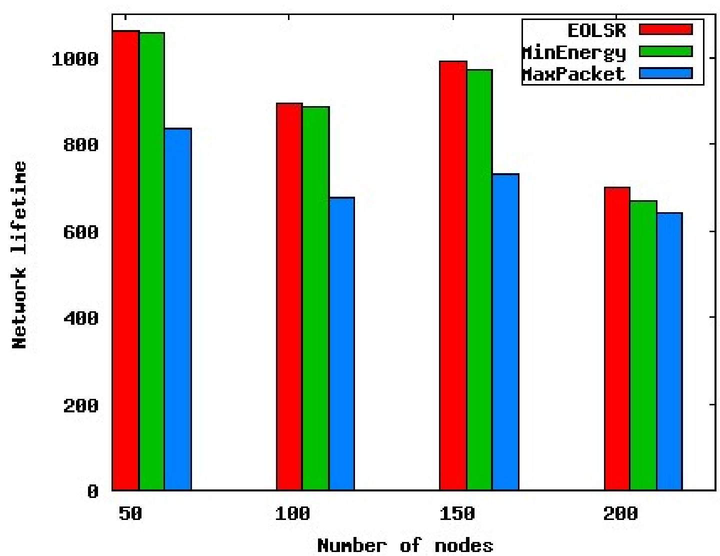

Figure 3.7: Comparison of the EOLSR network lifetime with $MinEnergy$ and $MaxPacket$.

does not take into account the energy cost when computing routes. As a consequence, the computed routes can be very long (in number of hops) and very expensive in energy consumption. $MinEnergy$ provides better results because it minimizes the energy consumed by each end-to-end transmission. $EOLSR$ outperforms both $MinEnergy$ and $MaxPacket$. This performance can be explained by the fact that $EOLSR$ takes into account the residual energy of intermediate nodes and the energy cost of the end-to-end packet transmission. Moreover it aims at sharing the load by changing the $EMPR$ selected set according to equation 3.9 ($\frac{E_R(new\_EMPR)-E_R(old\_EMPR)}{E_R(old\_EMPR)} > ThresholdEMPREnergy$).

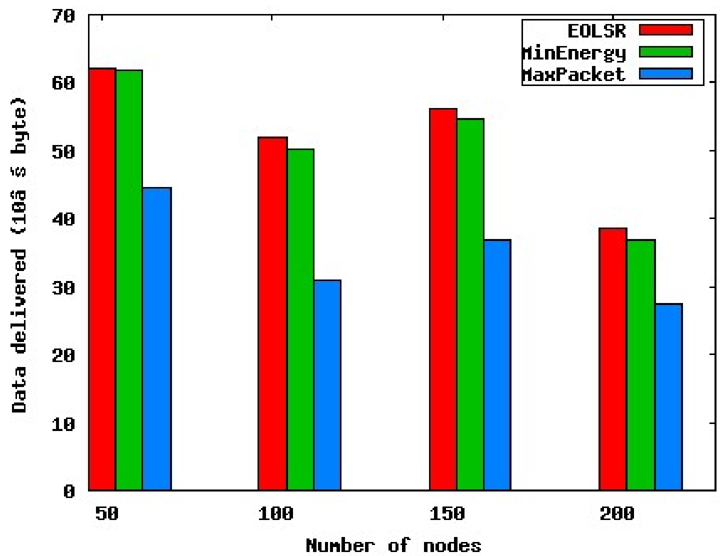

Figure 3.8: Comparison of the EOLSR data delivered with $MinEnergy$ and $MaxPacket$.

Figure 3.8 presents the data delivered in the network. Similarly, we can remark that $EOLSR$ outperforms both $MinEnergy$ and $MaxPacket$. Thus, the increase in the network lifetime is accompanied by an increase in the data delivered.

3.4.2 Comparative performance evaluation of $EOLSR$ with multipath routing strategies

Another strategy that can be used to spare energy is to share the load among the nodes. Multipath routing has been introduced for that purpose. In this section, we study two types of multipath routing: with different nodes called DN and with different links called DL.

3.4.2.1 Presentation of DN and DL

In multipath source routing, the source is in charge of computing several paths for each of its flow. As previously said, it has been shown that maintaining two paths is generally sufficient. The idea is for each flow packet, the source selects one of the paths with a probability inversely proportional to its cost. The selected path will be encapsulated in the packet. Hence, each visited node forwards the packet to the next hop on the path selected by the source. We distinguish two variants:

- **two paths with disjoint links**, denoted DL. The source of the flow computes a first path minimizing the cost, with Dijkstra algorithm. It then removes all the links used to compute the second path minimizing the cost with this new topology. Notice however that a same node (see node 2 in Figure 3.9) can belong to both paths and then deplete early.

- **two paths with disjoint intermediate nodes**, denoted DN. As previously, the source of the flow computes a first path minimizing the cost, with Dijkstra algorithm. It then removes all the intermediate nodes used and then computes the second path minimizing the cost with this new topology.

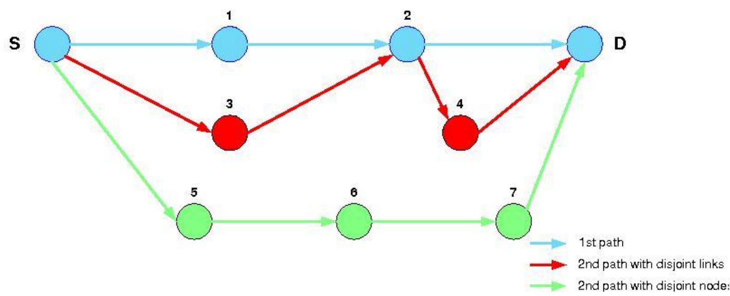

Figure 3.9: Multipath routing

3.4.2.2 Simulation parameters

Different parameters of simulation are listed in Table 3.4.

	Simulation parameter	Value
Configuration	Number of nodes	50-200
	Density	10
	Bandwidth	2Mbps
	Transmission range	250m
	Interference range	500m
Energy	Initial energy	uniformly distributed in [20J, 60J]
	P_{tans}	0.56Watt
	P_{recv}	0.16Watt
Traffic	Number of flows	30
	Throughput	16Kbps
	Packet size	512 bytes

Table 3.4: Parameters used for simulation.

Results reported in this section are the average of 10 simulations. In these series of simulation, we do not take into account the overhead induced by the different routing protocols. The network lifetime is defined as time of first network partition.

3.4.2.3 Comparative performance evaluation

In this second series of experiments, we compare the three energy efficient routing strategies $EOLSR$, DN and DL with regard to the network lifetime, when the number of nodes ranges from 50 to 200, the network density is set to 10. The $OLSR$ routing protocol that does never take energy into account and always selects the shortest path (i.e.; the path with the minimum hop number), is used as a reference. Simulation results are illustrated in Figure 3.10.

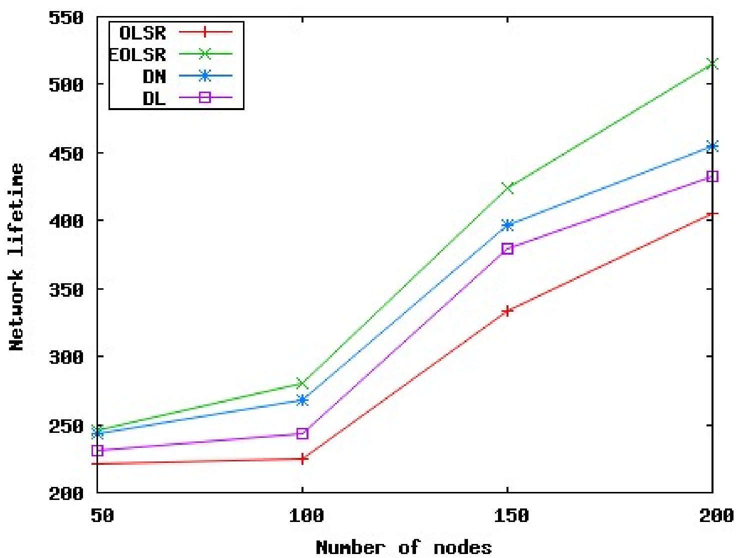

Figure 3.10: Comparison of network lifetime with $EOLSR$, DN and DL versus $OLSR$

As expected, $OLSR$ provides the smallest network lifetime. This shows that the selection of the shortest path is not sufficient to save energy. Concerning the two 2-path source routing strategies, DN provides better results than DL. This is not surprising insofar as energy is dissipated per nodes and not per wireless link. Hence, DL that allows common nodes in the two paths can exhaust the energy of these common nodes more quickly. The main conclusion of these experiments is that $EOLSR$ significantly outperforms DN and DL whatever the number of nodes. $EOLSR$ prolongs the network lifetime of 25% compared with $OLSR$ for a network of 200 nodes. Notice that in the same conditions, DN prolongs the network lifetime of only 10%. Indeed, the two paths chosen by the source of the flow are used for all flow packets independently of the residual energy of these nodes. So the intermediates nodes exhaust their energy more quickly.

Notice that the selection of an energy efficient routing protocol maximizing network lifetime, would provide no advantage to the application, if this increase in network lifetime was not followed by an increase in the amount of user data delivered. That is why, in this third series of experiments, we evaluate the delivery rate with the three routing strategies $EOLSR$, DN and DL, when the number of nodes ranges from 50 to 200. This delivery rate is compared with this provided by $OLSR$. Results are given in Figure 3.11. As previously, $EOLSR$ provides the best delivery rate, followed by DN, DL and finally $OLSR$. In conclusion, $EOLSR$ allows a real benefit to the application by delivering a higher amount of data with the same initial energy.

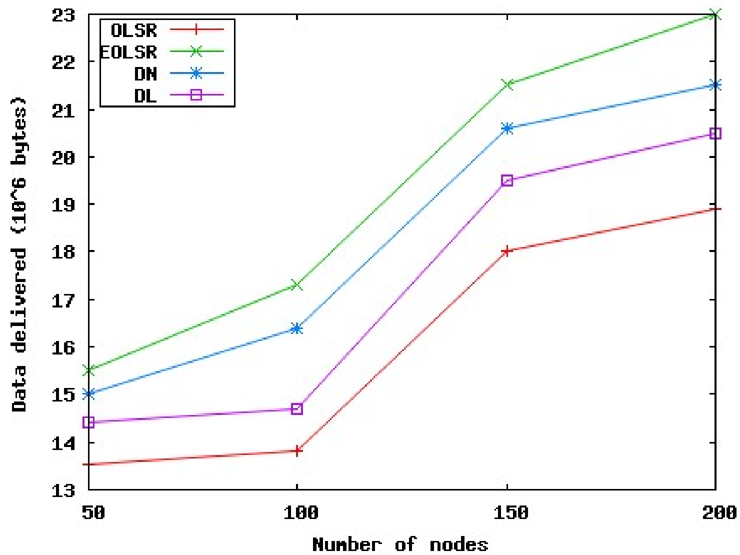

Figure 3.11: Comparison of delivery rate with $EOLSR$, DN and DL versus OLSR.

3.4.3 Energy consumption distribution

We now focus on how a node uses its energy. We consider the simulation parameters given in subsection 3.4.2.2 and study the distribution of energy consumption. In other words, we will quantify the energy dissipated in transmitting, receiving, overhearing, idle listening and interference during network lifetime while using the $EOLSR$ routing protocol. The initial energy is initialized at 100Joules.

These results show that the highest energy cumulative consumptions are due to the Idle and Interference states. An energy aware routing protocol can limit the energy spent in Interferences, as shown in Figure 3.12.

With regard to $OLSR$ (see Figure 2.1 in Section 2.7, Chapter 2), Figures 3.12 shows an improvement: less energy is lost in interferences. However, the least important part of energy is dissipated in the Receive and Transmit states that are the only states that matter from the application point of view. Too much energy is spent in the Idle state. To save energy dissipated in this state the only solution is to make nodes sleep: nodes that have nothing to transmit and receive can enter the Sleep state. This type of solution will be studied in the next chapter.

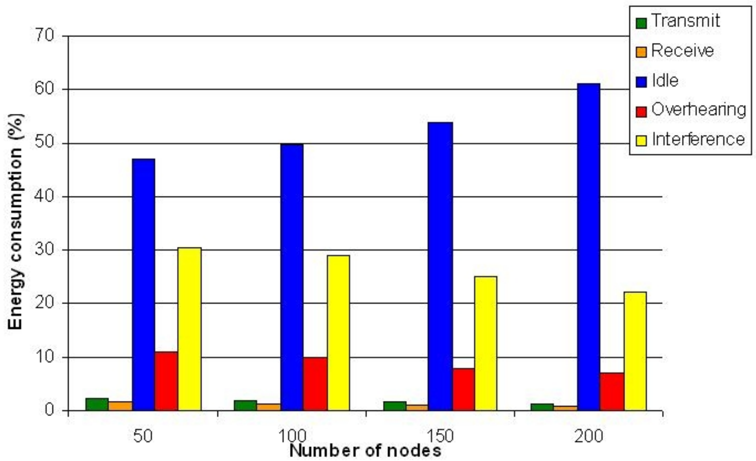

Figure 3.12: Distribution of node energy consumption without sleeping state.

In wireless sensor networks, there are many applications where the number of potential destinations is limited and known in advance by the application. This is the case of data gathering applications for example. For such types of application, we propose to optimize the EOLSR routing protocol. The idea is that each node in the network maintains only a route to each potential destination. This can be achieved with a cooperation between the application layer and the routing layer. In the next section, we will present this optimization and its advantages.

3.5 *EOLSR* for data gathering applications

In some applications, like data gathering, some nodes are strategic, like sink nodes. The other nodes need only to know a route to reach these strategic nodes. More generally, in wireless sensor networks, we can distinguish three types of data gathering application, according to their type of reporting:

- on-demand report: sensor nodes send their data in response to an explicit request from the sink.
- event-driven report: sensor nodes do not send data unless a certain event occurs.
- periodic report: sensor nodes send data periodically to the sink.

In all these cases, a sensor node should know a route to reach the sink.
The idea is to optimize the routing protocol for that the potential destinations are known. In this kind of application, all data must be sent to the sink nodes. Hence, instead of building a route from each node to any other node in the network, the idea is that each node must compute a route to reach each sink.

3.5.1 Maintaining routes only to strategic nodes

In the presence of strategic nodes, we propose to maintain routes only to strategic nodes. The EOLSR protocol is simplified as follows.

- Only strategic nodes periodically broadcast a TC message. This message contains (i) the TC origin that is a strategic node, (ii) a sequence number set by the origin node, (iii) the energy cost ($costEnergy$) that represents the energy dissipated to reach the strategic node (set to zero by the TC origin) and (iv) the predecessor from the sender of this message on the route to the TC origin.

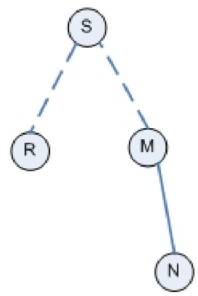

Figure 3.13: Building route to reach the strategic node.

- When any node N receives from node M, a TC message originated from node S (see Figure 3.13),
  - if the received information has a higher sequence number than the locally maintained one, let $d(N, S)$ designs the distance on number of hops between N and S
    - case $d(N, S) = 1$: node N updates its routing table with:
      - $nextHop(S) = N$, where $nextHop(S)$ is the next hop to reach S.
      - $costEnergy = E_{trans} + n * E_{recv}$, where $costEnergy$ is the energy cost to reach S and n represents the number of neighbors of N.
    - case $d(N, S) = 2$: if M is EMPR of N to reach S, node N updates its routing table with:
      - $nextHop(S) = M$,
      - $costEnergy = receivedcostEnergy + E_{trans} + n * E_{recv}$,
    - default: if M is EMPR of N to reach the predecessor of M,
      - if the received $costEnergy + E_{trans} + n * E_{recv} <$ local $costEnergy$ node N updates its routing table with: $nextHop(S) = M$ and $costEnergy = receivedcostEnergy + E_{trans} + n * E_{recv}$,
      - else N discards the message.

- * if (the message has not been discarded, N is EMPR of M, N uses M to reach S and it is the first forward of this message), N forwards the TC after having updated the $costEnergy$ and set the predecessor field to M.
  - else N discards the message.

This forwarding rule ensures that the propagation of TCs is done only by going away from the strategic node. Furthermore, it allows a node N to locally repair a broken link. Let us consider the following example depicted in Figure 3.14 where node F uses node D to reach S and the link between F and D is broken, F will try to reach B its 2-hop predecessor on the route to S. Knowing its 2-hop neighborhood, F will look for a 1-hop neighbor able to reach B; if it finds one, the route can be locally repaired.

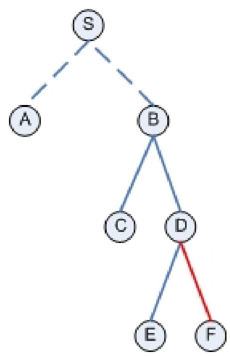

Figure 3.14: Local repair routes to reach the strategic node.

3.5.1.1 Differences between the proposed optimization and AODV a reactive routing protocol

The principle of AODV [112] is the following: when a source needs to communicate with another node, it broadcasts a route discovery message. Each node receiving this message, if it knows a route to this destination, it replies indicating the route to reach the destination. Otherwise, it broadcasts the route discovery message. Finally the destination replies with the route reply message. Our proposed algorithm differs from AODV in the following points:

- First, our protocol can be considered as a destination triggered routing algorithm. It is the destination that takes the initiative of sending the TC message, whereas in AODV, it is the source that triggers the route discovery algorithm.
- Unlike the route discovery of AODV, an optimized broadcast is used: only EMPRs nodes forward the message generated by the sink.

- Our solution allows all nodes in the networks to build a route to reach the sink. Indeed, the TC message sent by the sink is to build a route from any source to this destination. Whereas the route discovery of AODV is sent by a source to build a route to specific destination.

In the following, we compare the performance evaluation given by the strategic mode of EOLSR applied to applications where the set of destination is known in advance and the generic mode of EOLSR where each node build a route to each other node in the network.

3.5.1.2 Comparison between the strategic and the generic modes

We consider different random topologies, each topology being characterized by a number of nodes and a density (i.e. average number of neighbors per node). The density is set to 10. The radio range is 250m and the interferences are limited to two times the transmission range. Simulation results are averaged on 10 runs.

We consider a network of 100 nodes and evaluate the impact of the number of strategic nodes on the number of TCs sent per EMPR node per TC period and the number of TCs received per node per TC period.

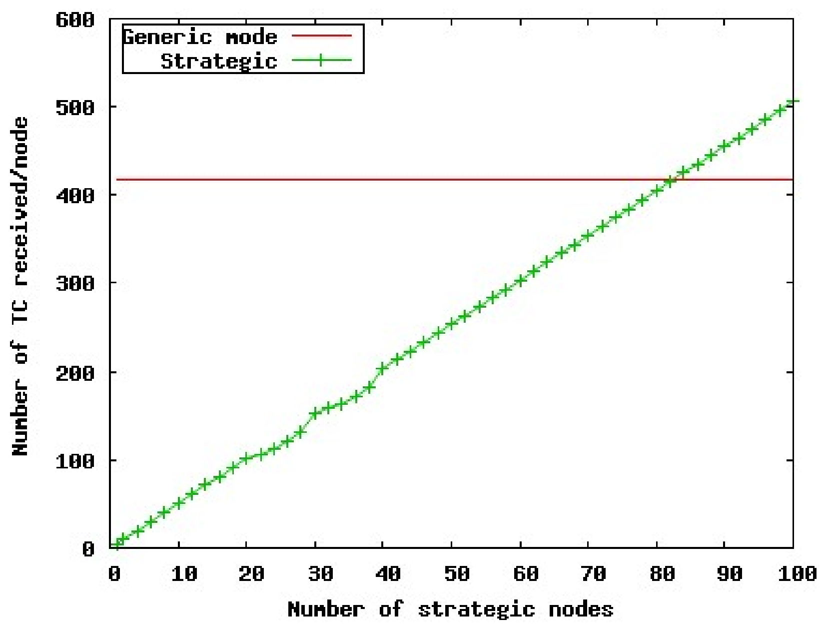

Figure 3.15: Number of TCs received per node per TC period in strategic and generic modes.

As shown by Figures 3.15 and 3.16, the proposed optimization considerably improves the EOLSR performance as long as the number of strategic nodes is smaller than the number of EMPRs (an average value of 82 in the considered simulations). The induced overhead is reduced (i) at the

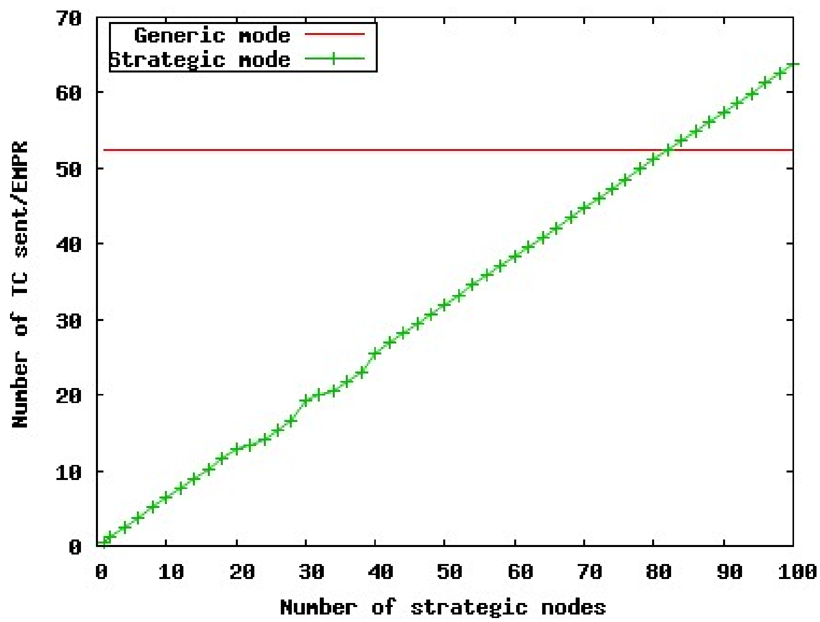

Figure 3.16: Number of TCs sent per MPR node per TC period in strategic and generic modes.

bandwidth level, (ii) at the processing level, (iii) at the energy level and (iv) at the memory level. Indeed, with this optimization, the routing table of a node is considerably reduced. It does no longer contain a route to any other node in the network, but only a route for its one-hop and two-hop neighbors and the strategic nodes.

3.5.2 Applicability of this optimization

The optimization given above can be applied to any data gathering application with either a periodic or an on-event report, as well as a general on-demand report (i.e. the sink requests all sensors to send their data). We now propose an extension for data gathering application with a specific on-demand report, where the sink solicits a specific sensor. In this latter case, the sink needs to know a route to reach this sensor. More generally, the sink maintains a route to each sensor it can individually poll. The proposed extension to the EOLSR protocol proceeds as follows:

- after having received a TC message from the sink, a sensor that can be polled by the sink waits a given duration denoted $Pause$ and then unicasts a $Join$ message to the node from which it received the TC, (in other words the next hop for the sink in its routing table). This message contains (i) the sink node as final destination, (ii) the sequence number of the TC sent by this sink, and (iii) for each descendant: its identifier and its sub-sequence number. This subsequence number is set to 0 for the first $Join$ message sent by this descendant to this sink with this sequence number; it is incremented at each transmission to another parent. The

Pause enables a node to aggregate the highest number of descendants in its *Join* message.

- the sink node builds its routing table: it inserts for each descendant the next hop, that is the sender node of the *Join* message with the highest sequence number and the highest subsequence number.

With this extension, the sink node knows how to reach each sensor node that it can poll.
More generally, the optimization given in Section 3.5.1 can be applied to any application where the number of potential destinations is smaller than the number of EMPRs and these nodes know in advance that will act as destinations. These strategic mode of EOLSR can be seen as an improvement brought by a cross layering with the application layer. We will see other possible improvement of EOLSR by means of cross layering in Chapter 6.

3.6 Conclusion

Wireless ad hoc and sensor networks are faced with the problem of energy efficiency in order to maximize network lifetime. The aim of this chapter was to extend the *OLSR* routing protocol to make it energy efficient. We have studied different variants of multipoint relay selection based on residual energy. The variant $M1E$ takes into account the energy dissipated in transmission and reception up to one-hop from the transmitter. $M1E$ presents the best tradeoff between the energy consumed and the overhead induced. Moreover, we recommend to keep the native MPR selection to optimize network flooding and to use the $M1E$ selection to build energy efficient routes.

We have also compared *EOLSR* with different routing strategies:

- *MinEnergy* and *MaxPacket*: hop-by-hop adaptive strategies selecting the route respectively with minimum energy cost and with the highest residual energy.
- *DL* and *DN*: two-path routing strategies with respectively different links and different nodes.

Simulation results show that *EOLSR* outperforms the other four strategies with regard to both network lifetime and delivery rate. This is due to the EMPR selection that avoids nodes with low residual energy. A high use of these EMPRs will reduce their energy. As a consequence, they will no longer be EMPRs. In other words, this EMPR selection contributes to uniform the residual energy of nodes. Moreover the choice of the route minimizing the end-to-end energy built with EMPRs improves network lifetime.
EOLSR operates in two modes depending on the application requirements:

- A generic mode where each node maintains a route to any other node in the network.
- A strategic mode where any non-strategic node maintains only a route to each strategic node. It is made possible by cross-layering between the application and the routing layers.

EOLSR improves the network lifetime. However, the energy consumption distribution results given in Section 3.4.3 show that the highest part of node energy is wasted in useless states (overhearing, interference and idle). To reduce the amount of this wasted energy, nodes should sleep as soon as they have no message to send or to receive. This must be accompanied by a node scheduling activity to avoid message losses and to keep the network functionality.

Chapter 4

Node activity scheduling

4.1 Introduction

In the previous chapter, we have shown the benefit brought by using energy efficient routing to improve network lifetime. However, the energy dissipated in useful states (receive and send) is small compared to the energy dissipated in the other states (idle, interference and overhearing). The idea is to reduce the energy dissipated in the idle state to improve the network lifetime. Moreover, as for a wireless node the state consuming the least energy is the sleeping state, allowing nodes to sleep is a good way to spare energy and avoid the idle state and the interferences. Nevertheless, when node turns off its radio and switches to sleeping state, it cannot sense, send or receive data. Therefore, sleeping nodes must not inhibit the application functionality and the network connectivity. That is why node activity scheduling is required. The goal is first to maximize network lifetime by allowing nodes to sleep (even router nodes), while ensuring end-to-end communication. Second, it is to optimize the use of network resources. Node activity scheduling consists in determining when each node in the network is active (it can send and receive messages) and when it is sleeping. The node activity scheduling that we propose is based on graph coloring algorithm. We then use a time slot algorithm based on node colors to schedule medium access.

In this chapter, first, we present a state of the art related to the two categories of graph coloring problem: vertex coloring and edge coloring. In the second section, we present our solution proposed to schedule node activity. This solution is named SERENA and consists of two distributed algorithms: a graph coloring algorithm and a slot assignment algorithm. First we focus on the graph coloring algorithm. We explain its principle and present some performance evaluation. Then we describe the slot assignment algorithm. Finally, we present a comparative evaluation of SERENA with classical TDMA and a dynamic variant of TDMA named USAP. In the third section, we show how SERENA can adapt to real wireless networks environments. In fact, real wireless network environments can cause problems that do not exist if we consider the ideal case (i.e. the case generally considered in graph theory) where all links are symmetric, messages are never lost, bandwidth capacity is very large, etc. Then we evaluate the impact of considering the wireless network in a real environment in the coloring algorithm and propose solution to solve the encountered problems. We end this chapter with a conclusion.

4.2 Graph coloring: state of the art

In graph theory, graph coloring is a special case of graph labeling; it is an assignment of labels, traditionally called colors, to elements of a graph subject to certain constraints. There are two categories of graph coloring:

- Vertex (or node) coloring: assigns a color to each vertex of the graph.
- Edge (or link) coloring: assigns a color to each edge of the graph.

In the following section we will define each type of graph coloring and present some related works.

4.2.1 Vertex coloring

Vertex coloring has received a lot of attention from researchers (see [65, 66, 67, 68, 69, 73]). It consists in coloring each vertex of the graph such that two adjacent vertices have not the same color and the number of colors used is minimum. This vertex coloring is named one-hop vertex coloring. This problem has been shown NP-complete in [63] for the general case, whereas graphs with maximum vertex degree less than four, and bipartite graphs can be colored in polynomial time.

The first one-hop graph coloring algorithms proposed were centralized (see [65, 66]). Among the greedy algorithms (i.e. no color backtracking), Dsatur, presented in [65], where the vertex with the highest number of already colored neighbor vertices is colored first, exhibits very good performances, even if it is not optimal. It is then followed by Largest First, where the node with the highest degree is colored first. Distributed one-hop graph coloring algorithms also exist. Some of them resort to randomization to select the color as for instance [71, 74] and [72]. The color selected by a node can be used only if it does not conflict with the colors chosen by its neighbors.

Other algorithms are strictly deterministic. The authors of [69] require that the results of the distributed algorithm and its centralized version be identical for any graph. This constraint is not required in the case of wireless ad hoc and sensor nodes. In [68], it is shown that in some network configurations, Distributed Largest First uses more colors than Largest First and the reverse is true in some other network configurations. This is because with Distributed Largest First, a node is allowed to keep its selected color, not only if it has the highest degree among its neighbors (as Largest First does), but also if there is no color conflict among its neighbors. Another approach consists in finding maximum independent sets and then coloring these sets independently, as in [70] and [69], because both problems are related [73].

Let n be the number of vertices and Δ the largest vertex degree. For one-hop graph coloring, the algorithm proposed in [73] runs in *O(log n)* rounds, and uses a number of colors close to Δ, whereas Distributed Largest First runs in $O(\Delta^2 \ log \ n)$ rounds [68].

4.2.2 Edge coloring

Edge coloring of a graph is an assignment of colors to the edges of the graph such that edges incident on the same vertex receive different colors. The edge coloring problems can be transformed into a vertex version. In fact, an edge coloring of a graph is just a vertex coloring of its line graph. Many solutions focus on this type of graph coloring to apply it to the wireless networks. These solutions are named link scheduling algorithm. The idea is to assign to each link between two neighbors a color (which corresponds to a time slot) in which these two neighbors can communicate.

Durand et al [82, 83] propose a distributed edge coloring algorithm for bipartite graph. The idea is such that each node randomly chooses one of its edge and assigns it a color. If no conflict is detected this edge keeps its color and is removed from the graph. Otherwise, (if there is a conflict) the node with the highest degree wins and so keeps the color of its edge. Marathe et al [85] conducted an experimental study of a distributed and randomized edge coloring algorithm and listed the scenarios where it fails. Initially each edge is given a palette, of $(1+\epsilon)\Delta$ colors, where Δ is the maximum degree of the network (graph). The computation proceeds in rounds. During each round, each uncolored edge, in parallel, first picks a tentative color at random from its palette. If no neighboring edge picks the same color, the color becomes final and the algorithm stops for that edge. Otherwise, the edge's palette is updated in the obvious way: the colors successfully used by the neighbors are removed and a new attempt is performed in the next round.

Gandham et al [86] proposed a distributed edge coloring algorithm that needs at most $\Delta + 1$ colors, where Δ is the maximum degree of the graph. Each color is mapped to a time slot and a direction of the transmission is determined to avoid the problem of the hidden and exposed terminal. Other solutions [87, 88] focused on the two-hop edge coloring wherein two edges are assigned different colors if they are adjacent or if there are connected by an edge.

The problem of assigning colors to edges for a general graph using the minimum number of colors is NP-complete [82].

4.2.3 Graph coloring applied to wireless sensor networks

Graph coloring has many advantages for the wireless networks. It permits:

- a better use of bandwidth: all nodes that have the same color can transmit simultaneously
    - without collision : collision avoidance results a bandwidth and energy gain.

- without interferences: nodes at two-hop of a transmitter are in sleeping mode if they are not at one hop of another transmitter.

- A better energy efficiency. Since sensor nodes have energy constraints, network protocols should be energy aware to maximize network lifetime. In this case, coloring algorithm permits to schedule node activity and so nodes that are neither transmitting nor receiving can sleep.

However, as we have seen in the previous section, there are two types of graph coloring and both can be applied to wireless networks to schedule node activity. The question now is which type is more adapted to our work?

4.2.3.1 Vertex coloring versus edge coloring

Edge coloring has many advantage compared to vertex coloring:

- Edge coloring can increase the concurrency of transmissions. In fact, if a color is assigned to a vertex then no one-hop or two-hop neighbors can use the same color in a collision free assignment. However, with edge coloring, two-hop neighbors can transmit or receive in the same slot by defining a sense for the edge. For example, with vertex coloring (see Figure 4.1), if node B sends a message to node A in its slot, no any one or two-hop neighbors can send message in this slot, in particular node C and D. However, with edge coloring (see Figure 4.2), links B A and C D have the same color. Hence B can communicate with A and C with D in the same slot and without collision.

Figure 4.1: Vertex coloring

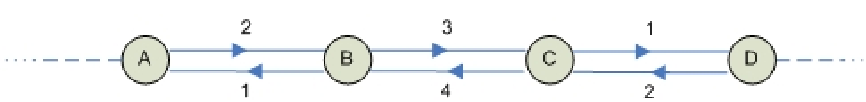

Figure 4.2: Edge coloring

- Vertex coloring restricts each node to transmit in at most a time slot in each frame, (if we use a bijective map between colors and slots) regardless of the number of one-hop neighbors that a node might have. However if colors are assigned to edges, nodes receive a time slot per neighbor. Hence, the bandwidth available to a node is proportional to its number of neighbors.

- Edge coloring can improve the energy conservation. In fact, when colors are assigned to nodes, all neighbors of a transmitting node must be awake during its slots to receive potential messages. Consequently, nodes waste energy in overhearing. whereas, if colors are assigned to edges then only the transmitter and the receiver must be awake. All other neighbors switch to sleeping mode.

However, the problem with the edge coloring is that:

- Edge coloring is oriented toward unicast communication. Therefore, broadcast communication is impossible. Nevertheless, broadcast messages as control messages are always needed to manage the network (e.g. Hello messages used for neighborhood discovery).
- The MAC layer must maintain a buffer for each one hop neighbor, it has messages to send.
- The number of colors used to color all the edges of the network can be very important compared to the number of colors used to color all nodes in the network. That is due to the number of edges which is much greater than the number of nodes in the network. Consequently, delay can increase with the number of colors.

In this work, broadcast messages are considered. They are important to manage the network and build the neighborhood. For this reason, we will use the vertex coloring approach. In the following, we use graph coloring to refer to vertex coloring.

4.2.3.2 Constraints for node coloring

Our solution for node activity scheduling supports unicast and broadcast transmissions. However, there are many other constraints that can be taken into account or not according to the user requirements. These constraints include:

- Types of unicast transmission:
  - generic mode: each node communicates with all its neighbors and messages can be destined to any node in the network. It can be applied to any types of application.
  - specific mode based on tree topologies: each node communicates only with its parent and its children in the tree. Messages are in general destined to the sink node (the root in the tree). It can be applied to data gathering applications.
- Immediate acknowledgement of unicast transmissions is:
  - required: each node must acknowledge immediately any unicast frame received correctly.
  - not required: acknowledgement is not required or it can be delayed.
- Minimizes the delay needed to collect data from all sensors:
  - in a tree topology applied to data gathering applications.

Our target is to design a solution that can be adapted to different constraints. In this chapter, we focus on generic mode valid for any type of application. We first consider, in Section 4.3, that an acknowledgement is not required or it can be delayed. Then we show how we can extend our solution to support immediate acknowledgement in the one hand and the real condition of the wireless network environment like unidirectional links in Section 4.4.1 on the other hand. In Section 4.5, we will see why color conflict can occur, how they can be detected and solved. In Chapter 5, we will present the specific mode dedicated to data gathering applications.

4.2.3.3 Efficiency of a distributed coloring algorithm

The coloring algorithm that we need must be distributed and localized: each node runs this algorithm to attribute itself a color based on local information. In fact a global information is very expensive in wireless networks.
The efficiency of a distributed coloring algorithm (edge or vertex coloring), [68, 73], can be evaluated by:

- the number of colors needed to color a graph G: closer this number to the chromatic number of G, more efficient the algorithm.
- its time complexity, expressed in the case of a distributed algorithm, by:
    1. the maximum number of rounds needed to color each node. A round is defined such that every node can:
        - send a message to all its one-hop neighbors,
        - receive the messages sent by them,
        - perform some local computation based on the information contained in the received messages.
    2. the number of messages exchanged to color a graph G: this number must be as small as possible.

In this work, we consider the problem of node activity scheduling in wireless ad hoc and sensor networks. Our goal is to improve the network lifetime with appropriate sleeping period while keeping the functionality and the connectivity of the network. To achieve that, we propose SERENA, an algorithm to SchEdule RoutEr Nodes Activity.

4.3 SERENA: Scheduling Router Node Activity

SERENA allows router nodes to sleep, while ensuring end-to-end communication in the wireless networks. It is a localized and decentralized algorithm assigning time slots to nodes. SERENA is based on two algorithms:

1. **two hop coloring algorithm**: it assigns a color to each node in the network such that two nodes that are one or two-hop neighbors have two different colors. Hence, a color is reused three-hop away. The number of colors and the complexity should be kept as small as possible.

2. **slot assignment algorithm**: it assigns a set of slots to each node in the network based on the coloring algorithm. These slots are used to transmit messages. Hence, each node must be awake in its slots to transmit its messages and in the slots of its one hop neighbors to receive their messages. It sleeps the remaining time.

These two algorithms are distributed and localized: only information about one and two-hop neighbors is needed.
In the following, we will first detail the two-hop coloring algorithm. Then we will present the slot assignment algorithm based on colors.

4.3.1 Two hop coloring algorithm

In wireless ad hoc and sensor networks, interferences are generally assumed to be limited to two hops. Hence, two transmitters at a distance strictly higher than two transmission range can simultaneously transmit. For this reason, at least two-hop coloring algorithm is needed to prevent collision. In this case, we consider the broadcast and unicast communications but not the immediate acknowledgment (i.e. the acknowledgement is not required or it can be differed).

As obtaining global information is very expensive in wireless ad hoc and sensor networks, the coloring algorithm must be localized and distributed: each node runs this algorithm based on its local information. To extend an algorithm of one-hop graph coloring to two-hop graph coloring can rise some difficulties. In fact, a node can communicate directly only with its one-hop neighbors. The information coming from its two-hop neighbors is received two rounds later. Let $\mathcal{N}^2(N)$ denote the set of nodes at a distance up to two hops from N, we can distinguish two classes of algorithms:

- the simplest ones, such as the extension of Largest First, are based on identical rounds, called decision rounds. In a round, a node sends a message to its one-hop neighbors, this message contains its color and the colors of its one hop neighbors. It receives the messages from its one hop neighbors and takes a decision if it has the highest priority. Its decision is based on decisions already taken by nodes in $\mathcal{N}^2(N)$ having a higher priority than N. The priority of a node is fixed and does not depend on the round.

- the more complex ones, such as Distributed Largest First and Dsatur, alternate proposal rounds and decision rounds. To propose a color, a node N must know all the decisions taken in the previous decision rounds, by nodes in $\mathcal{N}^2(N)$. To decide, a node N must know all the decisions taken in the previous decision rounds, and all the proposals made in the previous proposal round, by nodes in $\mathcal{N}^2(N)$.

The solution we propose belongs to the first class of algorithms that is simpler to implement and requires less messages, as illustrated by the comparative performance evaluation in Section 4.3.1.3.

The question is how to assign one color to each node such that the following requirements are met:

- two distinct nodes at a distance up to two hops have distinct colors,
- the number of colors used is as small as possible,
- the algorithm is distributed and localized to limit the overhead.

We represent the network by a graph $G(V, E)$, where V is the set of vertices which corresponds to nodes in the network and E is the set of edges which corresponds to links in the network. Two nodes A and B are said one-hop neighbors, in short neighbors, if and only if there exists a bidirectional link between them. Two nodes A and B are said two-hop neighbors if and only if there exists a third node C such that (1) A and C are one-hop neighbors, (2) B and C are one-hop neighbors and (3) A and B are not one-hop neighbors.

4.3.1.1 Two-hop coloring algorithm principle

The idea of our algorithm is to attribute to each node a priority. This priority defines the order in which the nodes will choose the colors as described in RC1.

Rule RC1: For any node N, if all nodes in $\mathcal{N}^2(N)$, with a higher priority than N, have a color assigned, N selects the color that is the smallest color unused in $\mathcal{N}^2(N)$.

In other words, if N has the highest priority among the not yet colored nodes in $\mathcal{N}^2(N)$, it takes the smallest color not used in $\mathcal{N}^2(N)$.
The set of colors in which the algorithm selects its colors, is assumed to be positive integers sorted in increasing order. The number of colors effectively used by SERENA coloring algorithm is not known a priori.
The set $\mathcal{N}^2(N)$ contains the one hop neighbors of node N and its two-hop neighbors. Node N builds its neighborhood up to two hops, by exchanging the following information with its one-hop neighbors:

Rule RC2: Each node N sends to its one-hop neighbors, a message containing its identifier, its priority, its color if already assigned, its list of neighbors their priority and the list of colors if already assigned.

4.3.1.2 Coloring algorithm

Procedure 2 Process(*Color*message)

if there is a change in the 1 or 2-hop neighborhood **then**
  update $\mathcal{N}^2(N)$
end if
if *priority*(N) has not yet been computed and the 1 and 2-hop neighborhoods are known **then**
  compute *priority*(N)
end if
maintain the priority and color of any node in $\mathcal{N}^2(N)$
if N is the node with the highest priority among the uncolored nodes in $\mathcal{N}^2(N)$ **then**
  N selects the smallest color unused in $\mathcal{N}^2(N)$
end if

Algorithm 3 Coloring algorithm

repeat
  repeat
    N broadcasts (1-hop) its *Color* message containing:
      - a sequence number *seq* incremented at each change in the *Color* message fields,
      - its identifier
      - its priority if already assigned
      - its color if already assigned
      - its list of one-hop neighbors with their identifier, their sequence number, their priority and color if already assigned
    N waits for the *Color* message of its 1-hop neighbors
    upon receipt of a *Color* **message,**
      N Process(*Color*message)
  until all one-hop neighbors have received the *Color* message of N with number *seq*
until all nodes in $\mathcal{N}^2(N)$ are colored

We propose two variants of our algorithm. These two variants depend on the priority assigned to nodes.

- The first one is based on **the node identifier**: the node with the smallest identifier receives the highest priority,
- The second one is based on **the cardinality of** $\mathcal{N}^2(N)$: the node with the highest cardinality, also called degree, receives the highest priority. If several nodes have the same cardinality, the node with the smallest identifier wins. Notice that this variant does not require additional knowledge, insofar as the set $\mathcal{N}^2(N)$ must be known in order to meet the requirements of two-hop coloring.

In the following section, we will evaluate these two variants of our coloring algorithm and compare them with Distributed Largest First (DLF).

4.3.1.3 Performance evaluation of the coloring algorithm

In this section, we compare the two variants of our algorithm with Distributed Largest First, DLF, extended to two hop coloring in terms of number of colors used and time required expressed in the number of rounds to get the coloring.

This distributed coloring algorithm does not require a synchronous system organized in rounds, even if for simplicity reasons, we use the term round to evaluate the time complexity of this algorithm. The information that must be known by a node is restricted to a distance of two hops. This meets the localization requirement. Simulations have been performed for different wireless networks, where the network density (the average number of neighbors per node) is fixed to 10. Nodes whose number varies from 50 to 200, are randomly distributed in the network area. Each result is the average of 10 simulations runs.

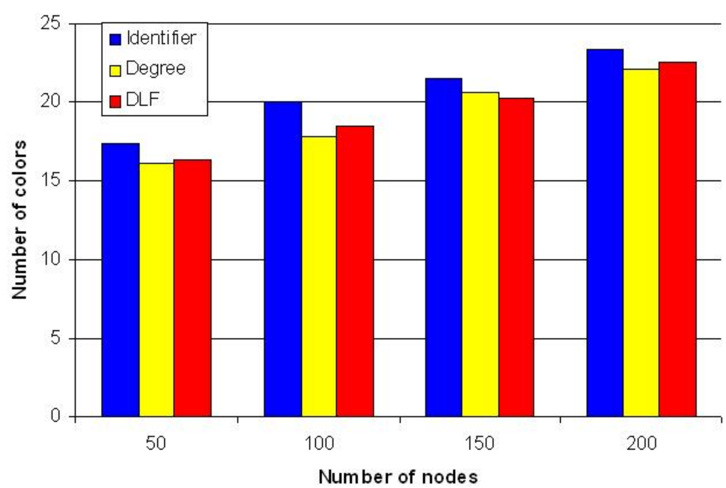

Figure 4.3: Number of colors used

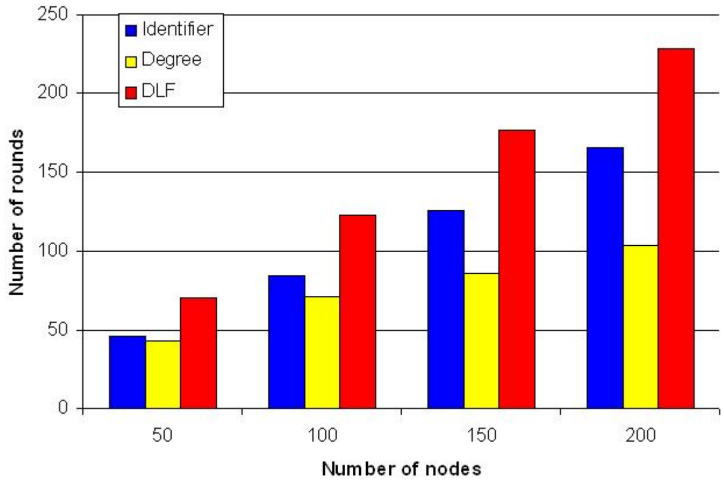

Figure 4.4: Number of rounds

Simulation results show that the variant with the maximum degree outperforms the variant with the smallest identifier, both in terms of number of colors used (see Figure 4.3) and time complexity expressed in number of rounds (see Figure 4.4). Intuitively, with the maximum degree variant, the information (i.e. the selected colors) is propagated more quickly in the network: more nodes know the selected colors and can decide. If we focus on the first node selecting its color, let N be this node. Two rounds later, the maximum number of nodes up to 2-hop from N knows that this color is chosen. At least one of them, with the maximum degree, selects its color, and so on.

Henceforth, **the priority of a node N in SERENA is set to the cardinality of $\mathcal{N}^2(N)$, denoted $|\mathcal{N}^2(N)|$.**
Compared with Distributed Largest First, DLF, our algorithm shows very good performance. Both algorithms use a similar number of colors, whereas the time complexity of our algorithm is significantly lower. Indeed, DLF alternates proposal rounds and decision rounds. To propose a color, a node N must know all the decisions taken in the previous decision rounds, by nodes in $\mathcal{N}^2(N)$. To decide, a node N must know all the decisions taken in the previous decision rounds, and all the proposals made in the previous proposal round, by nodes in $\mathcal{N}^2(N)$.

We now study the impact of network density on the numbers of colors and rounds used by DLF and our algorithm. We consider a network of 100 nodes, with a node density varying from 5 to 20. Simulation results are averaged over 10 simulations and illustrated in Figures 4.5 and 4.6 for the number of colors and the number of rounds, respectively.

With regard to these Figures, our algorithm provides an excellent performance, both in terms of:

- colors: it uses the smallest number of colors for densities 5 and 10 and is very close to the smallest number for density 20.

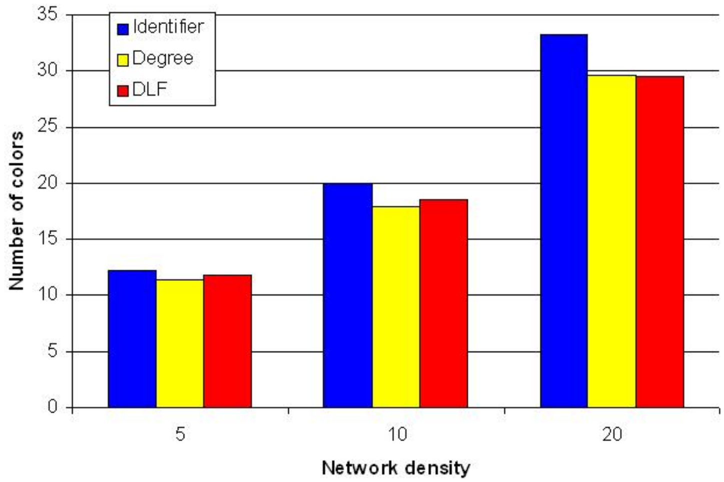

Figure 4.5: Number of colors used

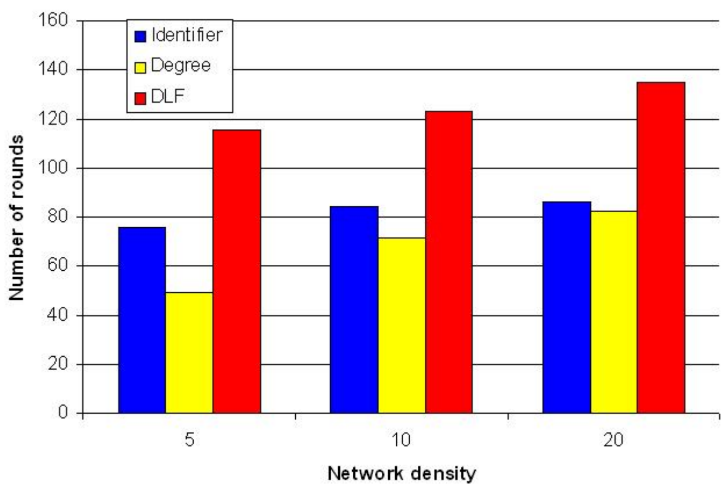

Figure 4.6: Number of rounds

- rounds: the difference between the two variants of our algorithm tends to vanish, when the density increases. For a density of 20, both variants have the same number of rounds. For all the densities studied, DLF exhibits the highest number of rounds, whereas our algorithm provides the smallest one. Hence, it provides a shorter convergence time, a very interesting property in wireless ad hoc and sensor networks where energy matters.

Compared with Figure 4.3, Figure 4.5 shows that the number of colors used by our coloring algorithm depends strongly on the network density and weakly on the number of nodes. This observation is also true to a lesser extent for the number of rounds. Let c denote the number of colors assigned by our algorithm, we have: $H1 + 1 \leq c \leq H2 + 1$, with $H1$ the maximum number of one hop neighbors and $H2$ the maximum number of neighbors up to two hops.

As soon as the coloring algorithm is achieved, the slot assignment algorithm is triggered. Then, the results of the two-hop coloring algorithm is used by the slot assignment algorithm to assign slots to each node. In the next section, we will present the slot assignment algorithm used by SERENA.

4.3.2 Slot assignment algorithm

A node N enters the time slot assignment algorithm as soon as it is colored as well as all nodes in $\mathcal{N}^2(N)$. The slot assignment mapping one color into a given number of time slots has the big advantage of simplicity. Furthermore, it allows each node to optimize locally the use of its time slot. Indeed, this time slot can be used indifferently for broadcast, unicast or both transmissions. The rule of node activity is that each node must be awake in its slots to transmit its messages and in slots assigned to its one hop neighbors to receive messages. There are two types of solutions for slot assignment:

- The simplest solution consists in assigning one time slot per color. Each node receives exactly one time slot per frame. The idea is to map each color to a unique time slot. This solution guarantees node fairness. It is adapted to uniform traffic. Hence, the node throughput would be equal to $Bandwidth/Size$, where $Size$ denotes the number of slots contained in the frame and $Bandwidth$ the network bandwidth.

- The second solution consists in assigning to each node several slots. This number of slots depends on the traffic rate of the node. This solution is more adapted in case of non-uniform traffic.

In the following, we will present the second type of solutions which consists in assigning slots taking into account the traffic rate of nodes.

4.3.2.1 Principle

To prevent starvation, the algorithm begins with assigning a time slot to each node in the network. This time slot is guaranteed and independent of the traffic rate of the node. Then others additional slots can be attributed to each node. The number of these additional slots depend on the traffic rate of the node. More precisely, executing slot assignment algorithm consists in applying the three following rules:

Rule RS1: Each node reserves a slot depending on its color. This slot is guaranteed to the node, whatever its traffic rate.

For simplicity, the node N, colored with color i, with $1 \leq i \leq c$, c being the number of colors used by the algorithm, receives slot i. This slot can be used to exchange control messages. We now focus on the additional slots granted to any node N. This number k' should be proportional to its traffic rate.

Rule RS2: If all nodes in $\mathcal{N}^2(N)$ with a higher priority than N have already selected their additional slots, node N selects its additional slots among the available slots. Their number is equal to k', with:

$$k' = \lfloor \frac{traffic(N)}{\sum_{i \in VisibleColor(N)} traffic(i)} * (Size - |VisibleColor(N)|) \rfloor.$$

where: $Size$ is the size of the frame; $|VisibleColor(N)|$ denotes the cardinal of the set of colors visible by N up to two hops; $traffic(N)$ is the bandwidth request of node N; it is computed from the traffic submitted by N on the last period and the traffic pending on N; $traffic(i)$ denotes the highest bandwidth request of nodes having color i up to two hops from N. Notice that several nodes in $\mathcal{N}^2(N)$ can have the same color: this is perfectly acceptable insofar as these nodes are not at a distance less than or equal to two hops. That is why the highest bandwidth request must be taken in the computation.
The available slots are the slots that are neither guaranteed, nor already granted to a visible color. If the slot assignment algorithm was centralized, slots would have been assigned per color. Hence, the number k' of slots would have been guaranteed to node N. Let us consider the following situation, illustrated in Figure 4.7.a, where color 1, used at node N_1, is reused three hops away, at node N_4. Node N_2 has color 2, whereas node N_3 has color 3. Let us assume a frame size of six slots. Three slots are allowed to node N_1, two slots to nodes N_2 and N_4, and one slot to node N_4. Colors receive their slots in increasing color order. Moreover, the number of slots allocated to color i is equal to the highest number of slots granted to a node with color i. We would have the slot assignment given in 4.7.a.

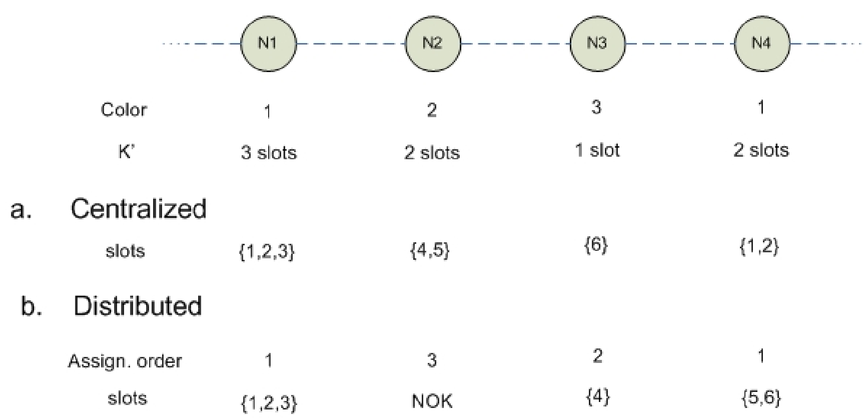

Figure 4.7: Distributed vs centralized slot assignment algorithm

In a distributed and localized slot assignment algorithm, each node locally assigns its slots. We can, then face the following situation, illustrated in Figure 4.7.b:

1. node N_1 selects slots 1, 2 and 3; meanwhile, node N_4 selects slots 5 and 6, because slots 1, 2 and 3 have already been attributed to neighbors with a higher priority than N_4;

2. node N_3 selects slot 4;

Finally, it is impossible for N_2 to select its two slots: there is no more available slot.

More generally, let us consider the following case where two nodes with a high priority (nodes N_1 and N_4 in the example), at a distance up to 4 hops reuse the same color and the union of the slots selected by one of these nodes is not included in the set of slots selected by the other (the set of slots $\{1, 2, 3\}$ is neither included in $\{5, 6\}$ and vice-versa). In such a case, nodes at a distance up to two hops from these two nodes (node N_2 in the example) will be unable to find k' available slots. To avoid this problem, SERENA proceeds as follows:

Rule RS3: If all nodes succeed in selecting their k' additional slots, the algorithm is over. Otherwise, let N a node that fails to assign its k' slots, it requisitions k slots among the requisitioning ones in $\mathcal{N}^2(N)$, with

$$k = \lfloor \frac{traffic(N)}{\sum\limits_{M \in \mathcal{N}^2(N)} traffic(M)} * (Size - |VisibleColor(N)|) \rfloor.$$

The requisitioning slots of a node are constituted by its additional $k' - k$ slots.
To be able to compute the bandwidth request associated with the visible colors, the message sent by any node must include the bandwidth request of this node and its neighbors. To know the slots already allocated up to two-hop, each node includes the list of slots allocated to itself and its one-hop neighbors.
Hence, each node has the guarantee to obtain at least $k + 1$ slots. It is possible to obtain $k' + 1$ slots, but without any guarantee: it depends on the configuration.

4.3.2.2 Adaptivity to traffic changes and optimizations

In order to adapt slot assignment to varying traffic rates, the slot assignment algorithm is run periodically. As we focus on wireless ad hoc and sensor networks with limited topology changes, we assume in this section that the color assigned to a node does not change. We will see in Section 4.5 how to deal with late node arrivals or node mobility.

We can notice three important features of this algorithm:

- no node starvation: each node is ensured to get at least one slot per frame. In other words, the minimum throughput guaranteed to a node is equal to $Bandwidth/size$, where $size$ denotes the number of slots composing the periodic frame.

- node fairness: if traffic is uniformly distributed and all nodes have the same degree, they all receive the same amount of time slots;

- adaptivity to varying traffic rates and non-uniform traffic patterns, where a few number of nodes submit a high part of network traffic.

Possible optimizations would consist in ordering the messages to transmit in a slot, in such a way that a node can detect the soonest possible that no message is destined to it. For instance, a node should:

- put the broadcast messages at the beginning of a slot,
- order the point-to-point messages by increasing destination identifier,
- switch to the sleeping state if it has no longer message to transmit in its slot. Its neighbors will detect the silence and switch to the sleeping state too.

Example of the execution of the slot assignment algorithm. In this section, we illustrate the behavior of our algorithm by a short example on a small network of 10 nodes. This network is illustrated in Figure 4.8, where the number besides the node identifier denotes the color assigned to this node by SERENA. The frame size is set to $2 * |\mathcal{N}^2(N)| = 16$ slots in this example.

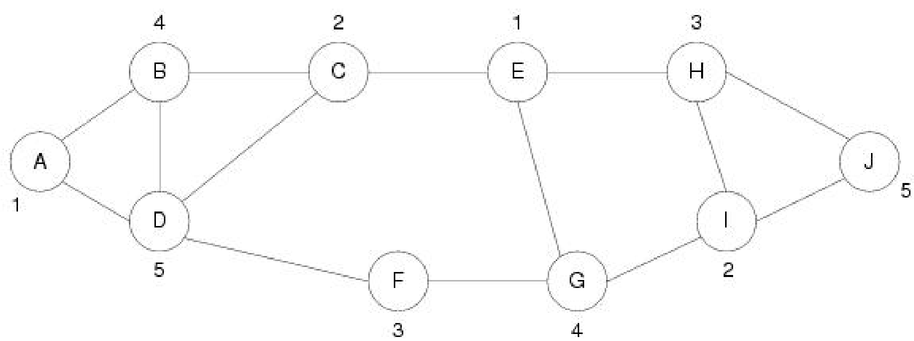

Figure 4.8: A wireless network and its coloring

The traffic for each node being given, the slot assignment algorithm computes the values of k and k', and assigns to each node a number of slots $(k' + 1)$ given in the last line of Table 4.1. As expected, node E which has the highest rate of traffic, receives the highest number of slots.

According to our algorithm, each node is awake during its slots and the slots attributed to its neighbors. The percentage of activity time is provided for each network node in Figure 4.9. The most active node is the node E that generates the highest traffic rate. Furthermore, its one-hop neighbors, C, G and H largely contribute to the network traffic, explaining why E has the highest activity. Meanwhile, nodes A and F are active at only 30%. Simulation results with larger networks (50 to 200 nodes) are reported in Section 4.3.4.

Node	A	B	C	D	E	F	G	H	I	J
Degree	4	5	7	6	8	7	7	5	5	4
Color	1	4	2	5	1	3	4	3	2	5
Traffic	10	20	30	10	50	10	30	50	30	10
k	0	1	1	0	2	0	1	3	2	0
k'	0	2	2	0	3	0	2	3	2	0
Slots	1	3	3	1	4	1	3	4	3	1

Table 4.1: Number of slots assigned per node

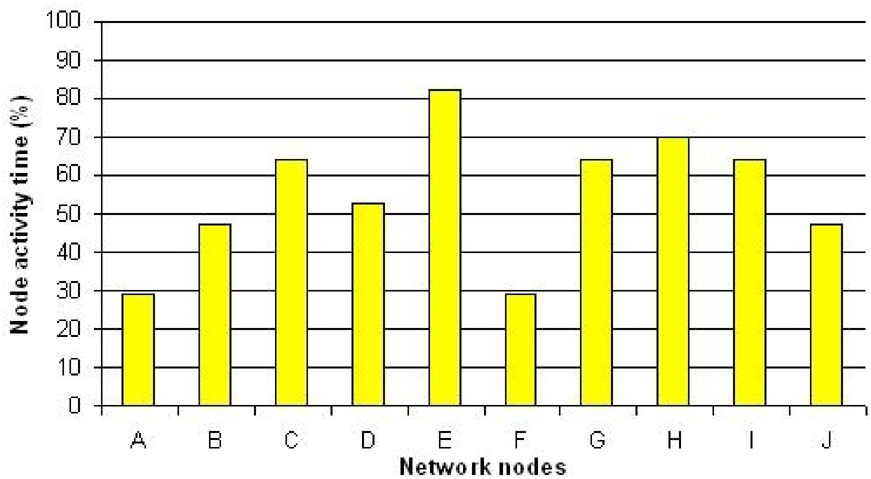

Figure 4.9: Node activity time

4.3.3 Message exchanged and information maintained in SERENA

4.3.3.1 Message exchanged

In this algorithm, only one message is exchanged named *Color* message. This message is broadcasted by a node to each one hop neighbors and never forwarded. This message is used to exchange information necessary to assign colors as well as slots. The *Color* message contains:

- the node identifier,
- its sequence number,
- its neighborhood up to two hops: $|\mathcal{N}^2(N)|$ (to compute priority, after calculating the priority, it never changes. This priority is unchanged with the set $\mathcal{N}^2(N)$),
- its color, if already assigned,

- the node traffic indication,
- the list of allocated slots and the indication whether they can be requisitioned or not,
- for any one-hop neighbor node:
  - the node identifier,
  - its sequence number,
  - $|\mathcal{N}^2(N)|$ (to compute priority),
  - its color, if already assigned,
  - the node traffic indication,
  - the list of allocated slots and the indication whether they can be requisitioned or not,

The *Color* message is periodically broadcast one hop. The sequence number of the sender is incremented at each change in the content of the *Color* message (except if the change is only in the sequence number of the one-hop neighbors). In order to tolerate message losses, a node N retransmits its *Color* messages until all its one hop neighbors have sent a *Color* message containing a sequence number for N equal to the current one. Notice that in case of topology changes, a link can be broken causing an update in the set $\mathcal{N}^2(N)$.

4.3.3.2 Information maintained by each node

Each node maintains the following information:

- its node identifier,
- its sequence number,
- its neighborhood up to two hops: $|\mathcal{N}^2(N)|$,
- its color, if already assigned,
- the node traffic indication,
- the list of allocated slots and the indication whether they can be requisitioned or not,
- for any one-hop neighbor node:
  - the node identifier,
  - its sequence number,
  - its neighborhood up to two-hop: $|\mathcal{N}^2(N)|$,
  - its color, if already assigned,
  - the node traffic indication,
  - the list of allocated slots and the indication whether they can be requisitioned or not,

- for any two-hop neighbor node:
  - the node identifier,
  - its neighborhood up to two-hop: $|\mathcal{N}^2(N)|$,
  - its color, if already assigned,
  - the node traffic indication,
  - the list of allocated slots and the indication whether they can be requisitioned or not,

Until now, we have presented the two algorithms of SERENA: the two hop coloring algorithm and the slot assignment algorithm and how it takes into account the traffic rate to assign slots. In the next section, we give a performance evaluation of SERENA in terms of the network lifetime, data delivered, spatial reuse and the distribution of the energy consumed in different states.

4.3.4 Performance evaluation of SERENA

	Simulation parameter	Value
Configuration	Number of nodes	50-200
	Density	10
	Bandwidth	2Mbps
	Transmission range	250m
	Interference range	500m
Energy	Initial energy	100Joules
	Transmit	1.3Watt
	Receive	0.9Watt
	Idle	0.74Watt
	Sleep	0.047Watt
Traffic	Number of flows	30
	Throughput	16Kbps
	Packet size	512 bytes
Frame	Number of slots	80
	Slot size	12ms

Table 4.2: Parameters used for simulation.

Simulations have been performed for different wireless networks. Different parameters of simulation are listed in the following table 4.2. User traffic consists of 30 flows, with randomly chosen sources and destinations. The frame consists of 80 slots. The frame size is equal to $2 * 4 * density$, where $4 * density$ is the average number of nodes up to two-hop. Simulation results are averaged on 10 simulation runs.

4.3.4.1 Network lifetime and user data delivered

In this section, we evaluate the benefits brought by the use of SERENA allowing router nodes to sleep. The routing protocol used is OLSR [34]. We compare the network lifetime obtained with and without SERENA. We quantify not only the network lifetime but also the amount of data delivered. Indeed, an increase in network lifetime is not sufficient if the volume of user data delivered is the same.

Figure 4.10 shows how the use of SERENA increases network lifetime. Network lifetime is defined as the time until a flow destination becomes unreachable (the network is no more connected). For 100 nodes, the lifetime increase reaches 420%. This result was expected. However, what really matters is the increase in the amount of user data delivered in the network, illustrated in Figure 4.11. For example, with 100 nodes, the increase in the volume of user data delivered reaches 404%.

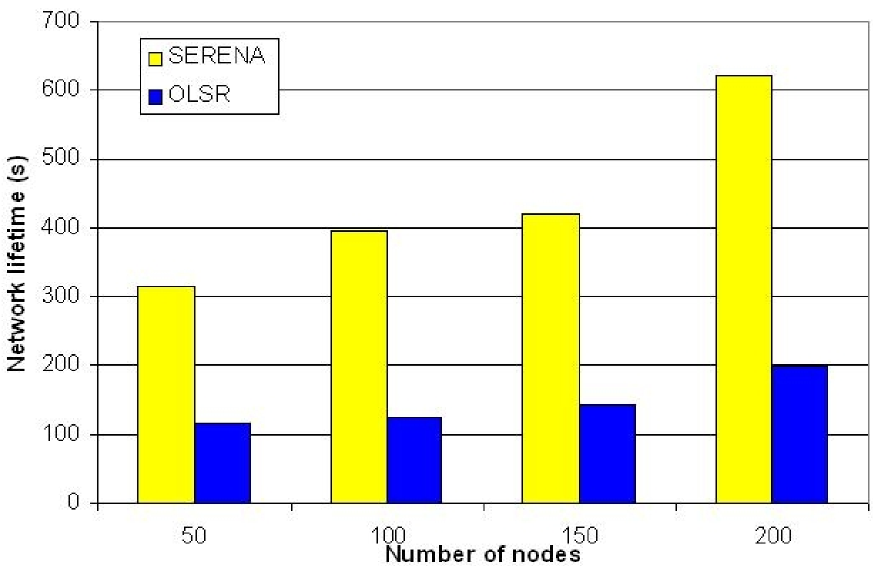

Figure 4.10: Impact of SERENA on network lifetime

We can identify two reasons for this high benefit:

- First, SERENA considerably reduces the time spent in the idle state. The idle state is the default state in wireless networks where sleeping is not allowed. The power corresponding to the idle state is about 16 times the power corresponding to the sleeping state. With SERENA, a node does not stay awake when neither it nor any neighbor are transmitting. Hence, less energy is dissipated uselessly.

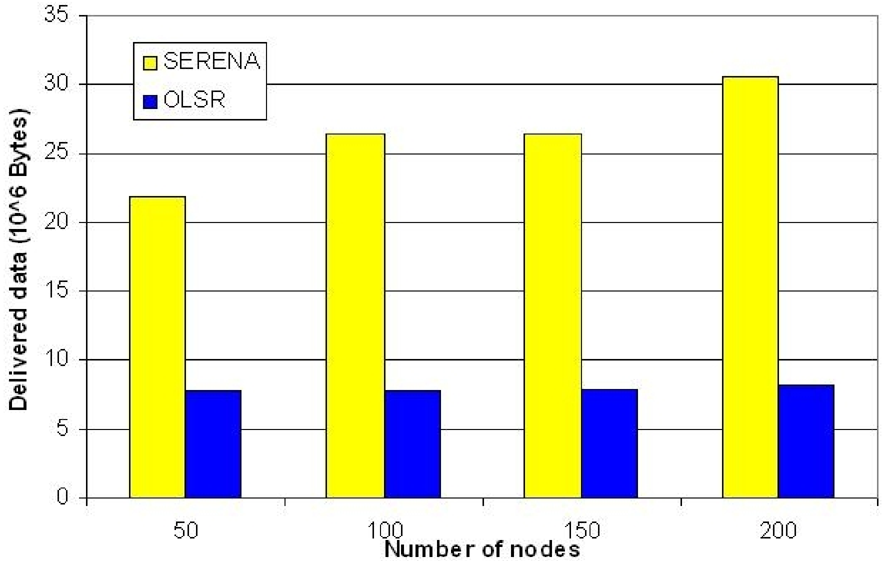

Figure 4.11: Impact of SERENA on delivered data

- Second, SERENA reduces the energy losses due to interferences. Indeed, a node is awake only during its slots and the slots assigned to its neighbors. Hence, SERENA tends to limit interferences to one-hop instead of two-hop. In wireless networks where sleeping is not allowed, a transmission dissipates energy on any node up to two hops from the sender.
- Third, the deterministic access to the medium with SERENA eliminates the energy wasted in the contention access period by the occurrence of collisions.

4.3.4.2 Slot reuse

We can notice that if all nodes have the same traffic, the average slot utilization is equal to:

$$average\ slot\ use = \frac{number\ of\ nodes}{number\ of\ colors}.$$

We now focus on the slot assignment algorithm and evaluate its performance by means of simulation. We consider a network of 50 nodes. The size of the node queue is set to 50. The other simulation parameters are unchanged. Traffic is not uniformly distributed over the nodes. We evaluate the number of slots obtained by a node with regard to its traffic rate.
Figure 4.12 depicts the number of slots assigned to ten nodes.

We notice that the nodes with the highest traffic rates (see nodes 3 and 6) receive the highest number of slots (7 and 11 slots respectively), whereas nodes with a small traffic rate (see nodes 1 and 9) receive few slots (3 slots for both of them). Simulation results show that as expected, SERENA assigns a slot number proportional to the traffic rate of the node.

Figure 4.13 depicts the slot reuse. It provides the number of slots shared by 5 nodes, down to 0 node. Two slots among the 80 are empty and 33 slots are reused by three nodes. Three quarter

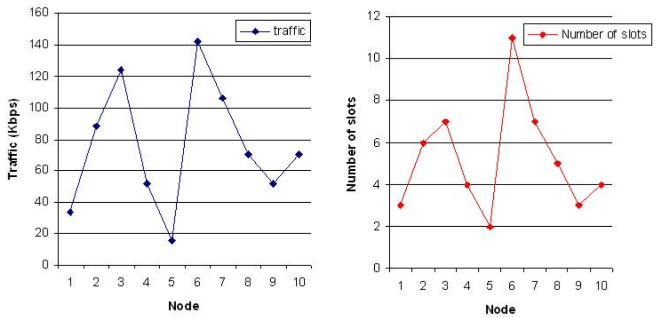

Figure 4.12: Slot assignment and traffic rate.

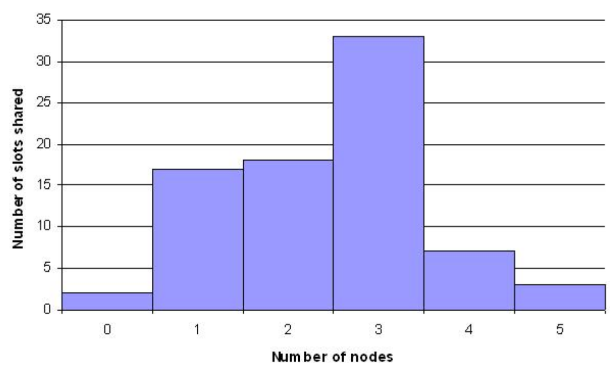

Figure 4.13: Slot reuse.

of slots are reused by several nodes. A slot is in average used by 2.43 nodes. This spatial reuse is obtained thanks to the SERENA two-hop coloring and slot assignment algorithms. Hence SERENA results in better network efficiency.

4.3.4.3 Distribution of node energy consumption

The distribution of node energy consumption is illustrated in Figure 4.14. The bar diagrams represent the energy dissipated in the Transmit, Receive, Idle, Overhearing and Interference states successively for different size of networks: 50, 100, 150 and 200 nodes.

Figure 4.14 shows that with SERENA, the energy dissipated in the idle state decreases to about 3% (instead of 50% as illustrated on Figure 2.1, in Section 2.7, in Chapter 2). This is the first benefit brought by SERENA. The energy cost due to interferences is about 3%. It was the second one without SERENA, with about 20%. This is explained by the fact that with SERENA, if neither the node nor its one-hop neighbors are transmitting, the node is sleeping. Hence, SERENA contributes to significantly reduce the interference phenomenon. This is the second benefit of SERENA. The energy cost due to overhearing can be reduced if messages are sorted according to their destination identifier.

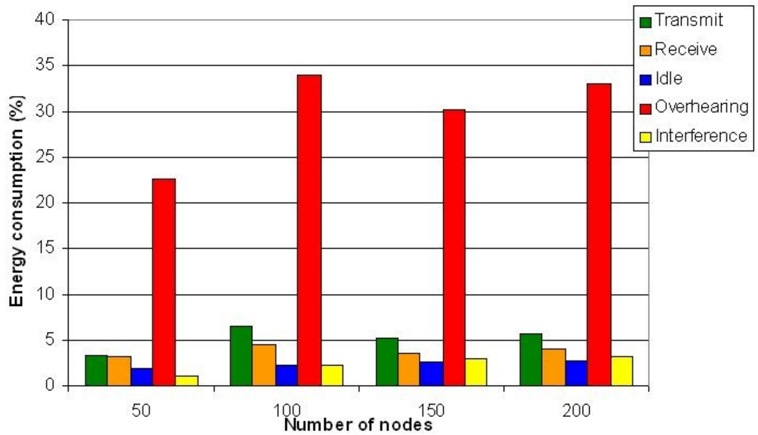

Figure 4.14: Distribution of energy consumption with SERENA.

4.3.5 Comparison with TDMA and USAP

In this section we will compare the performances of SERENA with classical TDMA and dynamic variant of TDMA named USAP (Unifying Slot Assignment Protocol) protocols.

4.3.5.1 TDMA and USAP

In its basic version, TDMA provides one transmission slot per node in the network. It provides a guaranteed access per cycle for every node and avoids collisions. However, it does not adapt to traffic variations. Many improvements have been proposed in order to take advantage of spatial reuse in wireless networks. Among them, USAP (Unifying Slot Assignment Protocol) [89, 90] has drawn a lot of attention.

USAP [89] is a distributed TDMA slot assignment protocol for mobile multihop packet radio networks. It allows any node N_i to select a slot for transmission that is unassigned in its neighborhood. A slot is unassigned from N_i point's of view if no one-hop neighbor of N_i transmits or receives during this slot. In our simulations, we consider an incremental allocation of slots. More precisely, a node requests an additional slot if the number of messages present in its queue is higher than a given threshold.

We will now compare the respective performance of USAP and SERENA in terms of network lifetime, data delivered and node throughput in different wireless ad hoc and sensor networks. These networks are characterized by their number of nodes and their density.

4.3.5.2 Comparative evaluation

In the simulations performed in this section, we use the simulation parameters presented in Table 3.2. The control overhead in SERENA and USAP is not taken into account.

Network lifetime and data delivered.

We focus on wireless networks of 100 nodes with density 10. The flow throughput ranges from 8kbps to 40kbps. We will compare the network lifetime and the amount of data delivered obtained by USAP and SERENA respectively.

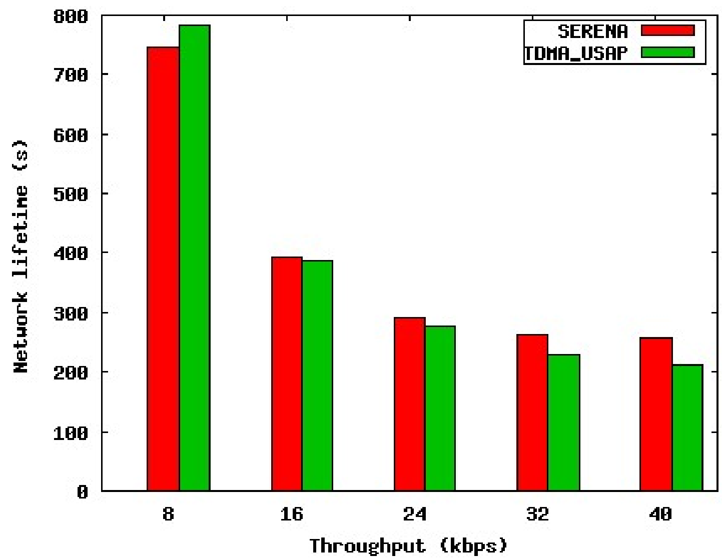

Figure 4.15: Network lifetime with SERENA and USAP.

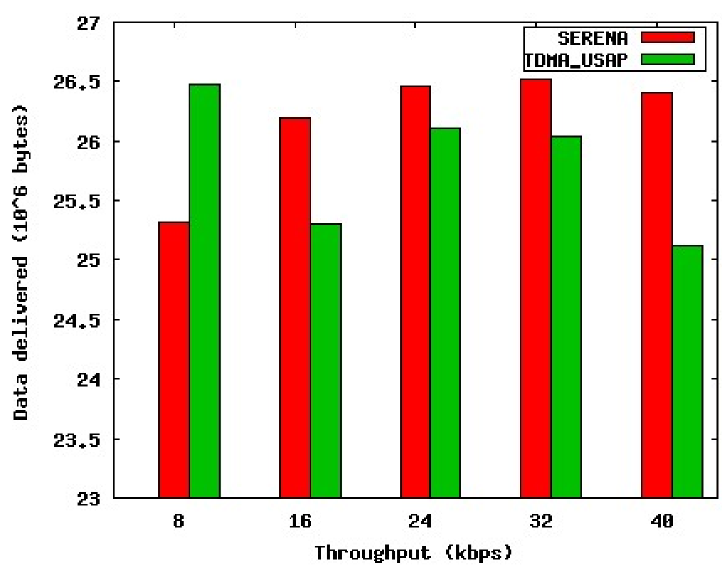

Figure 4.16: Amount of user data delivered with SERENA and USAP.

Figure 4.16 and Figure 4.15 depict the network lifetime and the amount of data delivered respectively in the network obtained as the flow throughput is increased. We notice that USAP outperforms SERENA when the throughput is equal to 8kbps. Indeed, if there is low load in the network, assigning one slot to each node is sufficient to transmit its traffic. That is why USAP is better than SERENA, which allocates more slots to nodes. However, when the throughput increases, we observe that SERENA become more efficient than USAP. We can explain that by the fact that in SERENA, assignment slots is based on:

- colors of nodes which provides an efficient spatial reuse,
- node traffic which allows to allocate the number of slots required to transmit its traffic.

Node throughput.
The throughput of a node can be evaluated as follows:

$$Throughput(N) = \frac{Number\ of\ slots\ allocated\ to\ N}{Total\ number\ of\ slots} \cdot Capacity\ of\ the\ medium.$$

We can compute the throughput obtained by each node as follows:
With classical TDMA, we have

$$Throughput_{TDMA}(N) = \frac{Capacity\ of\ the\ medium}{Number\ of\ nodes}.$$

With SERENA, the maximum throughput of a node N is given by

$$Throughput_{SERENA}(N) = \frac{k' + 1}{Size} \cdot Capacity\ of\ the\ medium,$$

where $Size$ denotes the frame size and k' is computed as presented in Section 4.3.2.
With USAP, the maximum throughput of a node N is given by

$$Throughput_{USAP}(N) = \frac{k'' + 1}{Size} \cdot Capacity\ of\ the\ medium,$$

where $Size$ denotes the frame size and k'' is determined by the queue size at the end of the slots allocated to the node considered, as presented in Section 4.3.5.1.

Therefore that both SERENA and USAP outperform classical TDMA because of their better spatial reuse. Moreover, SERENA better adapts to traffic requirements. Indeed, the slot assignment of SERENA takes advantage of the colors visible by a node (i.e. colors granted to the one-hop or two-hop neighbors) and shares the slots according to the sum on the number of visible colors, of the maximum slot number granted to each visible color. This is illustrated in Figure 4.17.

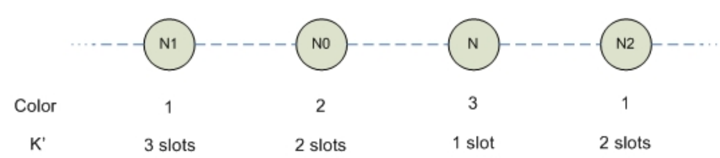

Figure 4.17: Slot assignment

To compute the slot assignment of node N, as represented on Figure 4.17, the maximum number of slots used by the two nodes N_1 and N_2, up to two-hop from N and colored with the same color,

Nodes	50	100	150	200
TDMA	40	20	13.33	10
USAP	50	50	50	50
SERENA	125	100	100	75

Table 4.3: Maximum throughput in Kbps obtained by a node.

is taken into account by SERENA (i.e. 3 slots). USAP takes the sum (i.e. 5 slots) if N_1 and N_2 have taken different slots.

Assuming that any node has always a message to send in the network and without routing, the maximum throughput that can be achieved in a network, with a frame Size equal to the number of nodes in classical TDMA and to $2 * 4 * density$ in USAP and SERENA, is given in Table 4.3. Each node is assumed to have $4 * density$ neighbors (including one-hop and two-hops). For SERENA, we assume the worst case, where a node sees all the colors.

We now consider a network of 100 nodes and density 10. Figure 4.18 compares the throughput obtained by nodes with SERENA and USAP respectively. For clarity, we represent the difference between the granted throughput and the requested throughput.

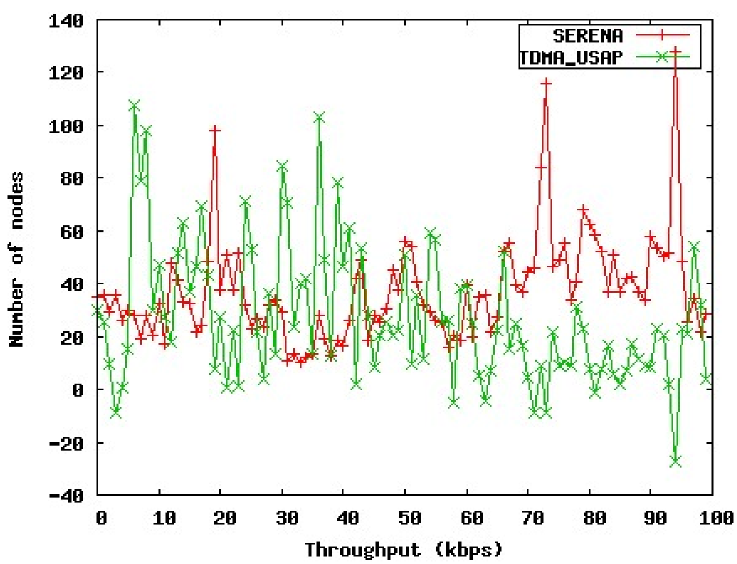

Figure 4.18: Node throughput obtained with SERENA and USAP.

We observe a better adaption of SERENA to traffic requirements. Hence, all nodes have at least their requested throughput. However, with USAP, some nodes have a granted throughput less than the requested one, which is not acceptable from the user point of view. These nodes are represented by negative value in Figure 4.18.

Throughout this section, we do not consider the constraints brought by wireless networks environment like topology changes, appearance of link, etc. This can be caused by the late arrival nodes, node mobility,etc. These constraints can cause problems for SERENA. In the next section, we will study, on SERENA coloring algorithm, the impact of considering a real wireless network environment. Then we will show how we can solve these problems caused by a real environment.

4.4 SERENA in a real wireless network environment

Until now, we have designed SERENA with the following network constraints:

- Unicast and broadcast transmissions.
- Generic type for unicast transmissions.
- Immediate acknowledgement is not requirement

However, the wireless nature of networks brings many constraints. In fact, when we consider unicast communication, an immediate acknowledgement is needed to confirm the correct receipt of the message and release the message from the node queue. Moreover, in wireless networks, links are not always bidirectional. Unidirectional links may be problematic in the coloring algorithm. In the following, we show how SERENA can be extended to support the immediate acknowledgement. Then, how SERENA solves the problems that can be caused by the real wireless network environment (unidirectional links, appearance of new links, etc.).

4.4.1 Support of immediate acknowledgement: Extension to three-hop coloring algorithm

4.4.1.1 Why three-hop coloring?

Generally, as we said in Section 4.3.1, in wireless ad hoc and sensor networks, interferences are assumed to be limited to two hops. Hence, two transmitters at a distance strictly higher than two transmission ranges can transmit simultaneously. However, if we consider unicast transmission, immediate acknowledgement is used to confirm the correct receipt of the message. In this case, two-hop coloring does not suffice to prevent collisions as we can see on node B in Figure 5.11, where nodes A and D three-hop away have the same color and transmit simultaneously. That is why three-hop coloring is used (color can be reused four-hop away).

Similar to two hop coloring algorithm presented in Section 4.3.1, the idea of three-hop coloring is to assign one color to each node in the network such that the following requirements are met:

- two nodes that are one, two or three-hop neighbors have distinct colors,
- the number of colors used is as small as possible,

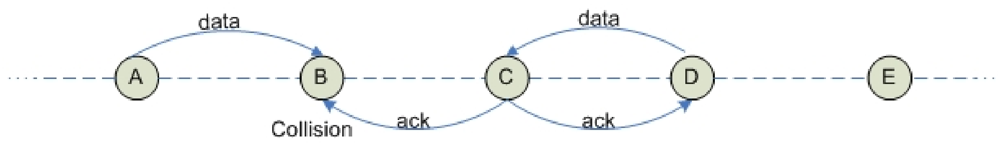

Figure 4.19: Collision with two-hop coloring and immediate acknowledgement.

- the complexity of the algorithm is minimized,
- the algorithm is distributed and localized: only information about one, two and three-hop neighbors is needed to run the algorithm.

In the following, we will first present three-hop coloring algorithm and give the performance evaluation of this algorithm in terms of number of colors and number of rounds. In the second step, we will study the impact of the unidirectional links on this coloring algorithm and how SERENA copes with this.

4.4.1.2 SERENA three-hop coloring algorithm

Two nodes A and B are said three-hop neighbors if and only if there exists a third node C and a fourth node D such that (1) A and C are one-hop neighbors, (2) C and D are one-hop neighbors, (3) D and B are one-hop neighbors and (4) A and B are neither one-hop nor two-hop neighbors. Let $\mathcal{N}^3(N)$ denote the set of one, two and three-hop neighbors of node N. First, each node in the network computes its priority. The priority of a node N is defined as the cardinality of $\mathcal{N}^3(N)$: the node with the highest cardinality receives the highest priority. If several nodes have the same cardinality, the node with the smallest identifier wins. This priority is unchanged with the change of the cardinality of $\mathcal{N}^3(N)$, once it has be computed. Nodes color themselves according to the order given by their priority.

More precisely, for any node N, if all nodes in $\mathcal{N}^3(N)$, with a higher priority than N, are already colored, N selects the smallest color unused in $\mathcal{N}^3(N)$.

Each node N sends to its one-hop neighbors, a message $Color$ containing its identifier, its priority, its color if already assigned, its list of one-hop neighbors, their priority and the list of colors already used by them, its list of two hop neighbors, their priority and the list of colors already used by them. Hence, each node N in the network can build its neighborhood up to three hops and knows all the colors already assigned to nodes in $\mathcal{N}^3(N)$.

4.4.1.3 Simulation parameters and evaluation criteria

Description of the topology generator

We use a custom-made generator to produce random realistic topologies. This generator is implemented by the LIMOS within the OCARI project [111].The generator works in the following way: it first builds a tree, places nodes in a rectangular area and computes the reception power for each pair of nodes. Notice that the coloring algorithm does not require the existence of a tree, but this tree is used to synchronize nodes in the MAC layer (see Section 6.6).

The tree is built according to topology parameters such as the maximum depth and the maximum number of children. Node placement is started at the root node of the tree. Each child of a node is placed at a random distance of its parent. Finally, we compute the power $pr_A(B)$ with which a node A can receive from a node B is computed using the ITU propagation model [84], for each pair (A, B). The ITU model suits realistic indoor deployment conditions by the use of a random variable.

The LIMOS team generated topologies with 25, 50, 100, 150 and 200 nodes and densities of 5, 10, 15 and 20 (plus or minus one). For each set of parameters, they created 100 topologies obtained by randomly varying the MAC layer parameters of the tree and the maximum distance of a child node to its parent. Once a topology is generated, they compute the set of good bidirectional links (that are above a threshold of -80 dBm) to ensure that the density is within the required interval, and that the topology is strongly connected. For density 5, the generator has failed to generate topology with 100, 150 and 200 nodes.

Note that the set of good links, *i.e.*, those which correspond to a reception power above -80 dBm, are not all bidirectional. This is due (i) to the random component of the propagation model which introduces a variability in the reception power and (ii) to the fact that we use a worst-case propagation model. Indeed, received power $pr_A(B)$ and $pr_B(A)$ are probably not independent in reality. However, as the variability of the propagation conditions on a link are difficult to estimate, we decided to have a worst-case model to prove the feasibility of the ability of SERENA to use unidirectional links.

4.4.1.4 Coloring results

In this section, we assume that all existing links are bidirectional and run three-hop coloring algorithm of SERENA in this environment. We have used 100 network topologies corresponding to a given node number and a given density. Each point depicted in Figures 4.20 and 4.21 is the average of 100 simulation runs. The performance of the coloring algorithm is depicted in Figure 4.20. We notice that the number of colors used to color all network nodes strongly depend on node density and more weakly on the node number. Figure 4.20 can be used to determine when the coloring algorithm improves network performance. Indeed, considering a given node number and a given node density, if the corresponding number of colors is less than the number of nodes, in other words this point is under the first bisectrix, then it is advantageous to color nodes. All generated configurations, except 25 nodes with a density ≥ 10, benefit from three-hop coloring.

The complexity of the algorithm is expressed by the number of rounds needed to color all network nodes. This number of rounds, depicted in Figure 4.21, is close to the number of nodes, whereas the worst theoretical case gives an upper bound of three times the number of nodes.

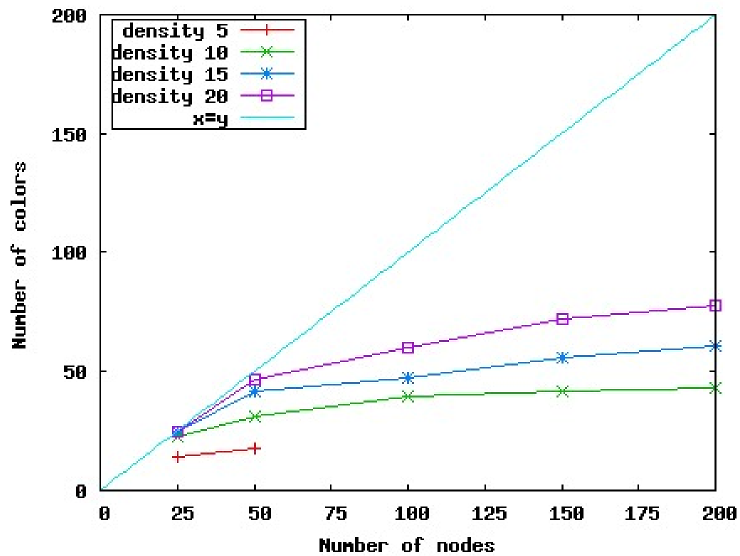

Figure 4.20: Number of colors used.

Now, we will consider the existence of unidirectional links and study their impact on the coloring

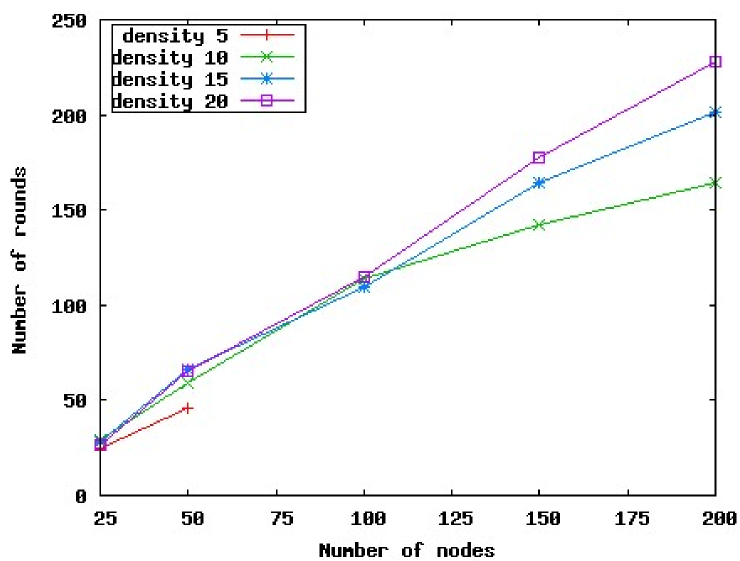

Figure 4.21: Number of rounds needed.

algorithm.

4.4.2 Coloring algorithm with unidirectional links

In this section, we consider the general case, which occurs in the real world, where some links are unidirectional. These links can cause color conflicts, as illustrated in Figures 4.22 to 4.25.

Definition of color conflict: *A color conflict occurs between two nodes having the same color when these nodes prevent each other or some neighbor destination to receive correctly the intended message because of a collision.*

Notice that in the absence of mobility, nodes that are one, two or three-hop neighbors never conflict by definition of the algorithm. Furthermore, this definition takes advantage of the capture effect and considers the only color conflicts where the intended destination is prevented to receive its destinated message.

The presence of unidirectional links can create color conflicts, between nodes that are at most three-hop away. We can prove by examining all possible cases that there are four types of conflict. Unidirectional links are represented by dotted lines , whereas bidirectional links are represented by a plan links. the node color is represented by an integer near the node.

- Type 1 color conflict: between a node A and a node B such that there exists an unidirectional link from A to B: B hears A that does not hear B, and A and B have chosen the same color. We can have a collision in B as depicted in Figures 4.22.

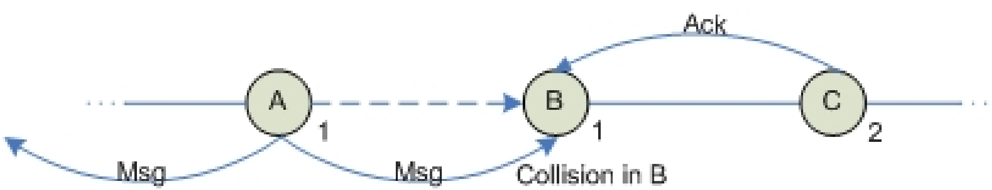

Figure 4.22: Type 1 color conflict between A and B causing a collision in B.

- Type 2 color conflict: between a node A and a node C such that there exists a bidirectional link between A and B, an unidirectional link from B to C: C hears B that does not hear C, and A and C have chosen the same color. We can have a collision in C as depicted in Figure 4.23;

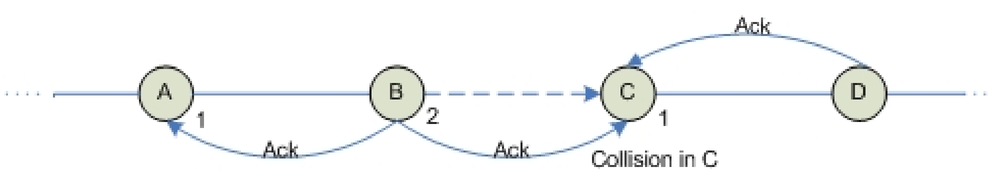

Figure 4.23: Type 2 color conflict between A and C causing a collision in C.

- Type 3 color conflict: between a node A and a node C such that there exists an unidirectional link from A to B: B hears A that does not hear B, a bidirectional link between B and C, and A and C have chosen the same color. We can have a collision in B as depicted in Figure 4.24;

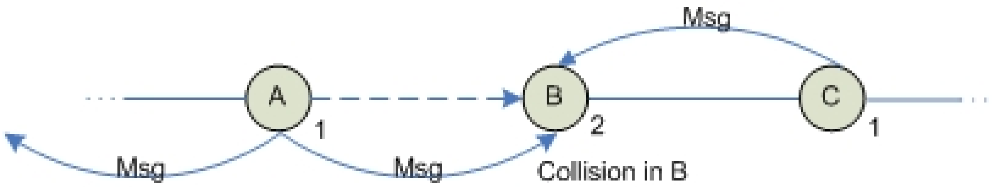

Figure 4.24: Type 3 color conflict between A and C causing a collision in B.

- Type 4 color conflict: between a node A and a node D such that there exists a bidirectional link between A and B, a bidirectional link between C and D, an unidirectional link from C to B, and A and D have chosen the same color. We can have a collision in B as depicted in Figure 4.25.

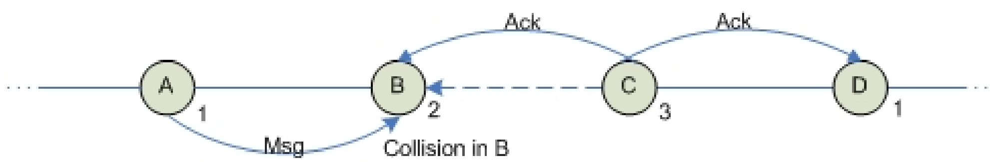

Figure 4.25: Type 4 color conflict between A and D causing a collision in B.

We also have the four symmetric cases of these four scenarios, where nodes A and its conflicting node exchange their roles. We now study the frequency of conflicts in the topologies generated according to Section 4.4.1.3. We also determine which conflict type is the most frequent.

We first characterize the environment in which the network operates. Besides the number of network nodes and the density, the average number of unidirectional links per node and the maximum number of unidirectional links per node give judicious information. Our topology generation tool generates an average rate of 20% of unidirectional links per node which represents a worst case in a real environment. It is interesting to study the behavior of our coloring algorithm in such adverse conditions. Simulation results are depicted in Figure 4.26. It turns out that for a given density, the number of conflicts increases with the number of nodes, as expected. Moreover, for a given node number, the number of conflicts decreases with the density. This can be explained by the fact that the probability for nodes to meet the conditions of scenarii 1 to 4, to be one, two or three-hop neighbors increases with the density. Hence, color conflicts are less frequent. If we focus on the type of conflict, we observe that the number of type 4 conflict is the most frequent followed by types 2 and 3. This can be explained by the fact that the nodes A and D of scenario 4 have a low probability to be 1, 2 or 3-hop neighbors, whereas nodes A and B of scenario 1 have a higher probability to be 2 or 3-hop neighbors. Hence a lower number of type 1 conflicts.

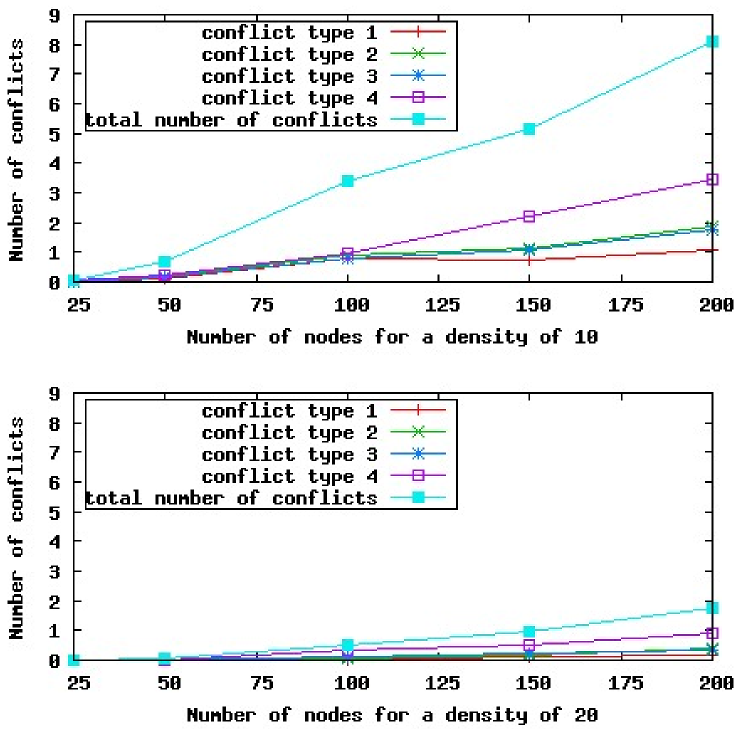

Figure 4.26: Number of conflicts per type.

4.4.2.1 Three variants of SERENA

In the presence of unidirectional links, the coloring algorithms can adopt one of the three following variants, ordered by increasing complexity:

- Variant 1: Ignore unidirectional links. This first variant is the simplest one. It is the one defined in Section 4.4.1.2. it is also the least consuming one in terms of bandwidth, processing power, memory and energy.

- Variant 2: Process information received from heard nodes. This second variant requires no additional field in the *Color* message. It only processes information broadcast by heard nodes and avoids to select a color already used by an heard node.

- Variant 3: Process and Transmit information received from heard nodes. This third variant requires additional fields in the *Color* message: the colors used by (i) the heard nodes, (ii) the neighbors of the heard nodes and (iii) the neighbors of nodes heard by one-hop neighbors.

From these definitions, it follows that variant 1 generates the highest number of conflicts, whereas variant 2 tries to avoid types 1 and 2 conflicts, and variant 3 all the four types of conflicts. Notice however that variant 3 avoids color conflicts only if node A colors itself before:

- node B in type 1 conflict,

- node C in types 2 and 3 conflicts,
- and node D in type 4 conflict.

In short, all these variants are unable to ensure the absence of color conflict in all cases.

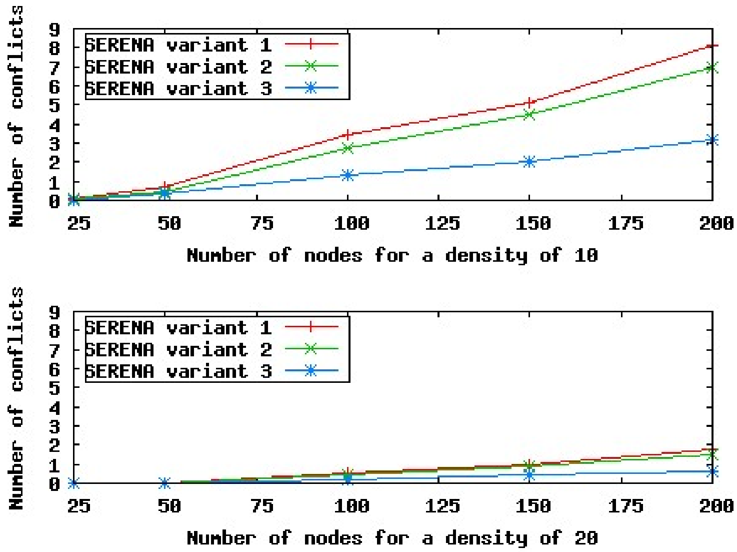

Figure 4.27: Number of conflicts per SERENA variant.

Simulation results are depicted on Figure 4.27 where the total number of conflicts and its distribution in the different types are given for each variant of SERENA. The upper figure concerns networks with density 10, whereas the lower figure concerns networks with density 20. As expected, variant 1 that ignores the existence of unidirectional links creates the highest number of conflicts. Variant 2 decreases this number by avoiding as much as possible types 1 and 2 conflicts. Variant 3 provides the smallest number of conflicts by trying to avoid all four conflicts types. None of them ensures zero color conflict. However, it is interesting to notice that the highest number of conflict does not exceed 8 in all cases, nevertheless we have introduced a high number of unidirectional links (20%per node). That is why color conflicts should be detected and solved. Furthermore, as the network bandwidth is limited, we choose the variant of SERENA that does not increase the message size and tries to avoid types 1 and 2 conflicts: variant 2.

4.5 Color conflict detection and resolution

4.5.1 Causes of a color conflict

The existence of unidirectional links is not the only possible cause of a color conflict. Mobility is another one. Because of mobility, nodes A and B that were not 1, 2 or 3-hop neighbors, can become

1, 2 or 3-hop neighbors either directly because of their own mobility or because of the mobility of some other nodes that creates new links making nodes A and B 1, 2 or 3-hop neighbors. Similarly, late arrivals can cause new links making two nodes with the same color 1, 2 or 3-hop neighbors. Hence, we can distinguish three new types of color conflicts caused by mobility or late arrivals:

- type 5 conflict: nodes A and B having the same color become one-hop neighbors, due to mobility;
- type 6 conflict: nodes A and B having the same color become two-hop neighbors, due to mobility or late arrivals;
- type 7 conflict: nodes A and B having the same color become three-hop neighbors, due to mobility or late arrivals;

As a result, color conflicts are unavoidable in a real wireless environment. That is why, we propose to detect and solve such conflicts.

4.5.2 Principles of detection and resolution of color conflict

In order to increase the reactivity to color conflicts, we propose to have the MAC layer detect and notify SERENA of any conflict. Indeed, the MAC layer can use a specific medium access mechanism to detect unexpected behaviors and report information concerning nodes in conflict to SERENA solves this conflict. The solution that we will use is proposed in a joint work with the LIMOS in OCARI project.

4.5.2.1 Description of TDMA/CA

Although SERENA reduces the possibility of color conflicts, they are still possible as explained in 4.5.1. For instance, such conflicts can result from node mobility or changes in the radio propagation conditions. In the following, we use the term *desired frame* to denote a frame a node is supposed to receive, and *undesired frame* any other frame.

The LIMOS team proposes an hybrid MAC layer mechanism called TDMA/CA that is able to detect color conflicts. TDMA/CA enhances the TDMA approach with CSMA/CA random backoffs. It is applied during all the slot dedicated to a color given by SERENA. It works as follows. Generally, nodes transmit their frames immediately. This can cause collisions, notably at the beginning of the slot as nodes are synchronized. Collisions are detected by the absence of acknowledgements. In this case, to desynchronize nodes, random backoffs are used.

However, not every color conflict results into a collision, even when two frames or more arrive at the receiver at the same time. In order for the reception of a frame to be lost, the sum of the received power of all the other frames has to be higher by a threshold than the received power of the desired frame. When it is, the frame is said to be captured. Thus, three cases can occur: (1) the *desired frame* can be captured, (2) the *undesired frame* can be captured, or (3) no frame is captured, which is the collision. Fig. 4.28 depicts how TDMA/CA behaves in each of these three cases. Note that in this example, the collision is of type 1, type 3 or type 4.

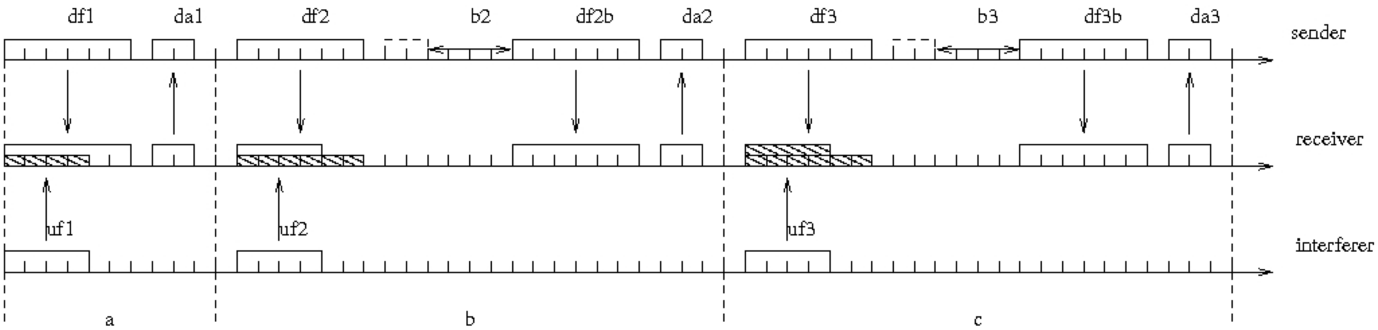

Figure 4.28: TDMA/CA behavior when (a) a desired frame is captured, (b) an undesired frame is captured or (c) a collision occurs.

Figure 4.28 shows the behavior of TDMA/CA for a sender node (upper line), a receiver (middle line) and an interfering node (bottom line). In the left part of the figure, the sender sends a desired frame to the receiver. At the same time, the interfering node sends an undesired frame. However, the desired frame is captured, and the receiver replies with an acknowledgement to the sender.

In the middle part of the figure, the sender sends a desired frame to the receiver while the interfering node sends an undesired frame. In this case, the undesired frame is captured. The receiver node can extract from this frame the information about the color conflict, and can report it to SERENA. Moreover, the sender which has not received the acknowledgment from the receiver delays its retransmission by a random backoff, and retransmits the frame.

Finally, in the right part of the figure, the desired frame sent by the sender and the undesired frame sent by the interfering node collides at the receiver. This happens when the reception power of the desired frame is close to the reception power of the undesired frame. As in the previous case, the sender has to delay its retransmission by a backoff. Notice that the receiver does not acknowledge undesired frames, as it is not supposed to be the destination of those frames. Also notice that the goal of the random backoff is to desynchronize the sender and the interfering node. If the interfering node heavily uses the channel, the backoffs drawn by the sender allow some of the undesired frames to reach the receiver. The receiver thus generates a color conflict notification, along with the sender as the number of retransmissions exceeds a threshold.

4.5.2.2 Benefits

- TDMA/CA allows us to use the simplest version of SERENA: any color conflict generated by SERENA can be detected locally.
- TDMA/CA approach allows collision avoidance: color conflicts can be tolerated as long as the number of collisions does not exceed a threshold.
- If there is no color conflict, TDMA/CA does not introduce any delay and acts in this sense like any other TDMA approach.
- Even if there is a color conflict, TDMA/CA does not notify SERENA as long as the desired frames can be captured.

4.5.2.3 Notification to SERENA

To notify SERENA, the MAC layer that uses TDMA/CA needs to know the addresses of the two nodes in conflict. This information can be obtained from undesired data frames or undesired acknowledgements in the MAC headers. Notice that this requires the MAC layer to operate in promiscuous mode, so that it does not reject the undesired frames it receives. SERENA is notified only when the number of retransmissions (which is contained in the MAC header of data frames and acknowledgements) exceeds a threshold.

Upon detection of a color conflict, the MAC layer locally notifies SERENA indicating the addresses of the two conflicting nodes. When a node N receives such an indication, two cases are possible:

- either N is a conflicting node, as for example in the color conflicts of types 1 or 2. If N has the smallest priority or the conflicting node is not a 1, 2 or 3-hop neighbor, then N selects another color. Otherwise N sends a $Conflict$ message to the other conflicting node inviting it to change its color.
- or N is not a conflicting node, as for example in the color conflicts of types 3 or 4. N sends a $Conflict$ message to the conflicting node that is either the only 1, 2 or 3-hop neighbor, or has the smallest priority.

The $Conflict$ message can be routed to reach its final destination, as in case of color conflict of type 7.

When a node must change its color, it selects the smallest color different from the previous one and not used in its neighborhood up to three hops.

4.6 Synchronization

The principle of SERENA is to assign to each node a time slot to send its messages without collision. Hence, the frame is divided into equal slots. This frame is repeated periodically. To know the beginning and the end of this frame a global synchronization is needed. In fact, each node must know exactly when the frame begins and when it must wake up to send messages (its slots) or to receive eventual messages (slots of its one hop-neighbors). The problem in a distributed network is that there are no global clock or common memory. Each node has an internal clock. For these reasons, a network synchronization must be developed. There are many researches in literature that aim at synchronizing wireless ad hoc and sensor networks. We can classify these solutions in two classes:

- hierarchical synchronization. We can site as example TPSN [92]. TPSN is a sender-receiver based synchronization that uses a tree to organize the network topology. All nodes will be synchronized with the root node. It is run in two phases: 1) a level discovery phase to build the tree, and 2) the synchronization phase to synchronize all nodes in the tree with reference to root clock. A network-wide time synchronization in sensor networks [94] is slightly different from TPSN. It is aimed at ensuring that synchronization accuracy does not degrade significantly as the number of nodes being deployed increases. The synchronization in the OCARI project belongs to this class of solution. In fact, the network has a tree topology. The PAN coordinator (the root of the tree) synchronizes all network nodes by means of its beacon that is broadcast multihop (more explanations are given in Section 6.6.2.3).

- non hierarchical synchronization. FTSP [93] was designed for large multi-hop networks. The root is elected dynamically and periodically reelected and is responsible for keeping the global time of the network. The receiving nodes will synchronize themselves to the root node and will organize in an ad hoc fashion to communicate the timing information amongst all nodes. The network structure is mesh type topology instead of a tree topology as in TPSN. RBS [91] exploits the broadcast nature of wireless communication medium. It uses receiver to receiver based synchronization. The idea is that a third party broadcasts a beacon to all the receivers. This beacon does not contain any timing information. However receivers use its arrival time to compare their clock. The timing is based on when the node receives the reference beacon.

With the synchronization, SERENA can be applied to any type of medium access (TDMA or CSMA/CA) providing time slots

4.7 Conclusion

The most efficient way to spare energy is to make node sleep, while ensuring network and application functionalities. In this chapter we presented SERENA. SERENA allows nodes to sleep. Any node (even router node) can sleep in all time slots except those allocated to it and its one-hop neighbors. SERENA is a localized and decentralized solution based on a coloring algorithm. Simulation results show the benefits brought by SERENA: network lifetime is improved and the amount of user messages delivered is increased proportionally to the network lifetime. Then, we have shown how it is important to consider wireless network topologies representative of real ones, when designing energy efficient protocols. More particularly, the presence of unidirectional links, as they exist in real environments, induces color conflicts in SERENA. We have identified the different types of color conflict caused by unidirectional links, late arrivals or mobility. We have studied different variants of SERENA whose complexity increases with the number of conflict types partly avoided. The chosen solution minimizes complexity. Since color conflicts are possible, we have proposed a cross-layer approach between SERENA and the MAC layer to detect and solve these conflicts.

Assigning times slot to each node using a color algorithm is a good way to improve the energy efficiency of the network and optimize its resources consumptions by benefitting from the spatial reuse. However, in some particular applications like data gathering allocating slots, without taking care in color assignment can lead to poor performance. For instance, the time needed to collect data can be equal to a number of cycles equal to the depth of the data gathering tree. Moreover the collected data correspond to a physical phenomenon that evolves with time. It is then essential to minimize the delay needed to collect these data from sensors generating them. Furthermore, ensuring that all data are gathered in a single polling cycle guarantees their time consistency. To achieve these goals and then maximize the advantage brought by coloring, coloring must take into account the data gathering tree in color selection. In the next chapter, we will present how SERENA can be adapted to this type of application.

Chapter 5

Node activity scheduling for data gathering applications

5.1 Motivations

In this chapter, we consider data gathering applications, where a sink, called root, collects information from sensors. **Our goal is to satisfy the requirements of such applications, while using network resources (bandwidth and energy) efficiently**. We can quote as examples of application requirements:

- Some applications are time critical. For example, alarms must be transmitted in a bounded delay to the sink. This bounded delay is usually equal to one cycle, where the cycle is the time interval in which all sensors have the opportunity to send data.
- Other applications require the delivery of consistent information to the sink, this information has been collected from individual sensor nodes. Since data collected in sensor networks are usually time varying, the time needed to collect fresh information from all sensors must be smaller than a given threshold depending on the application considered to ensure information consistency. For example, temperature and pressure should be measured in the same cycle. Moreover, it is essential to guarantee that data are successfully received by the sink.

In such scenarios, to guarantee a time slot to each sensor for sending data is insufficient to meet the requirement of the applications quoted above. Efficient data gathering scheduling is needed, which specifies how the data collected at each individual node is transmitted to the sink within the shortest period of time.

In wireless sensor networks, resources have limited capacity: for example, a bandwidth of 250 kbps for ZigBee [95]. As a consequence, these resources must be used efficiently, taking advantage of parallelism as much as possible. Spatial reuse will increase the amount of bandwidth available to the application and reduce the time needed to collect data from all sensor nodes. The idea is to adapt SERENA to the special application of collect/dissemination of data. In fact, without care, node activity scheduling can increase the delay needed to collect information from a sensor. In the worst case, a number of cycles equal to the distance, in hop number, to the sink is needed to reach the sink. To avoid this phenomenon, a node must transmit its data before its parent in the data gathering tree.

In this chapter, we focus on wireless sensor networks where nodes are organized in a tree rooted at the sink. This tree can result from the node attachment procedure like in IEEE 802.15.4 and ZigBee. It can also be built dynamically as the result for instance of an optimized broadcast from the root.

Our goal is to improve: the energy efficiency, the throughput and the end-to-end delays guaranteed to a data gathering application running on a wireless sensor network, taking advantage of slots assigned to nodes according to a node coloring algorithm.

This chapter is organized as follows. In Section 5.2, we present how we can adapt SERENA's three-hop coloring to data gathering application in case of tree topology. We present in Section 5.3 an optimization of this algorithm. We evaluate its performances and compare them with those of TDMA and TDMA-ASAP [96] for different configurations of wireless sensor networks. We compare the performance obtained by a tree colored by our algorithm and those given in [102] for a classical ZigBee tree [59]. We assume that the input from each sensor node is bounded by a linear arrival curve and evaluate the benefits brought by our algorithm in terms of the amount of bandwidth reserved to each router in the contention free period, the buffer size and finally the maximum end-to-end delay. Finally, we conclude this chapter in Section 5.6.

5.2 SERENA adaptation to data gathering applications

5.2.1 Principle

To maximize the network resources provided to a data gathering application, we take advantage of spatial reuse by means of coloring algorithm. However, this is not sufficient to decrease the end-to-end delays. The activities of nodes must be properly ordered in order to allow any sensor data to reach the root of the tree in a single cycle. The idea is to adapt SERENA coloring algorithm to data gathering applications by determining a color assignment order such that no child is scheduled after its parent in an upstream communication. This coloring algorithm consists in finding the smallest number of colors to color any node in the network, provided that any node N has a color higher than its parent in the data gathering tree.

We adopt the following notations. Let $\mathcal{T}$ be any data gathering tree and N any node in this tree. We denote *cycle* the time between two successive beacons of the root of the tree.

- $identifier(N)$: the identifier of node N,

- $Desc(N)$ denote the set of descendants of N in $\mathcal{T}$. By descendant, we mean the children, children of the children and so on... Let $|Desc(N)|$ denote the cardinal of this set.
- $color(N)$: the color of node N. A color is identified by a natural integer (starting with zero).
- $parent(N)$: the parent of N in $\mathcal{T}$.
- $\mathcal{N}^3(N)$ denote the neighborhood up to 3 hops from N. This set contains the 1-hop, 2-hop and 3-hop neighbors of N.

To achieve our goal presented above, we change the priority of nodes presented in SERENA to be adapted to communications according to a tree: For any node N, we say that N has a higher priority than node $N' \in \mathcal{N}^3(N)$ if and only if:

- either N' meets $|Desc(N)| > |Desc(N')|$
- or ($|Desc(N)| = |Desc(N')|$ and $identifier(N) < identifier(N')$).

We keep the same rule RC1 present in SERENA and change RC2 as follows:

- **Rule RC1:** any node N colors itself if and only if all nodes in $\mathcal{N}^3(N)$ having a higher priority than N are already colored.
- **Rule RC2:** node N takes the smallest color available in $\mathcal{N}^3(N)$ strictly higher than the color used by its parent.

It follows that:

- the first node to color is the root of the tree. It takes color 0,
- each node has a color strictly higher than the color of its parent and than the color of all its ascendants in the tree.

Figure 5.1 depicts the colors assigned to 15 nodes called A,B...O building a tree rooted at node A. Eight colors are needed. Hence, eight slots are sufficient (if we assign a time slot to each color) instead of 15 for classical TDMA. The same color can be used by several nodes (e.g; color 6 is used by nodes K, L, M and O).

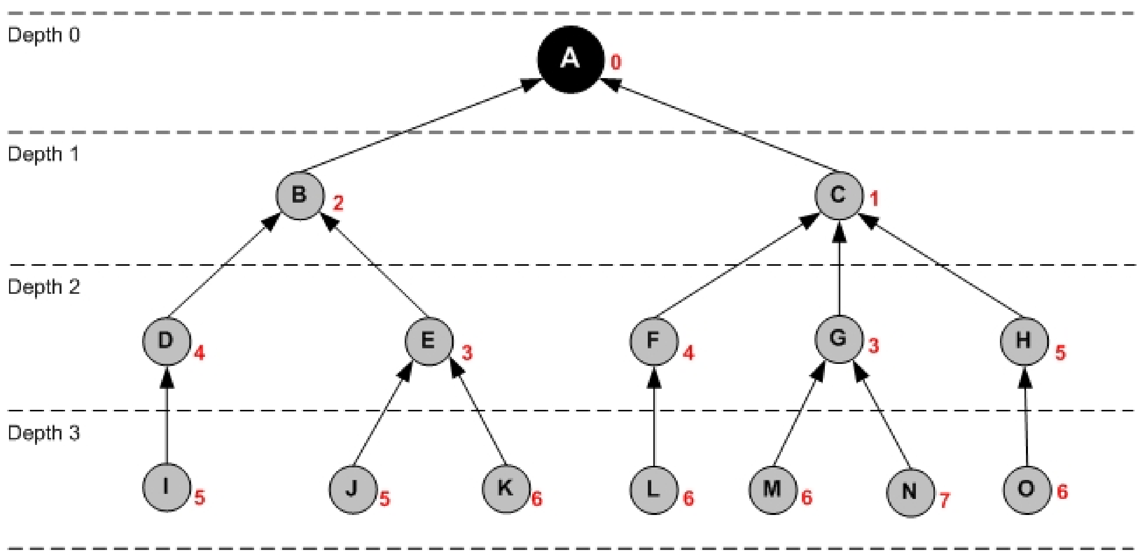

Figure 5.1: Colors assigned to a tree of 15 nodes.

Figure 5.2 depicts the slot assignment. For instance, nodes K , L, M and O that have the color 6 transmit in the same slot, numbered 6. Notice that each child transmits before its parent. Hence, in the cycle of eight slots illustrated by this figure, any sensor can send its value to the sink, assuming data aggregation.

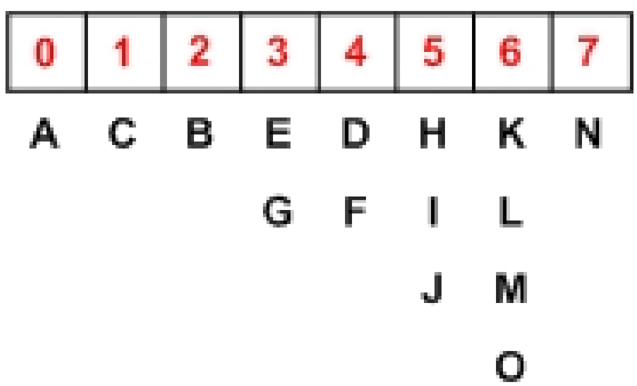

Figure 5.2: Slots assignment.

5.2.2 Messages exchanged Information maintained by each node

In this algorithm, two messages are exchanged:

- **the** *Color* **message** defined in SERENA: it is the only message needed to color all network nodes. This message contains the same fields presented in Section 4.3.3.1. Only the information concerning the size of the neighborhood up to three-hop must be replaced by the number of descendant in the tree to compute the new priority.

The *Color* message is broadcast one-hop. The sequence number of the sender is incremented at each change in the fields of the *Color* message, except the sequence number fields. In order to tolerate message losses, a node N retransmits its *Color* messages until all its one-hop neighbors have sent a *Color* message containing a sequence number for N equal to the current one.

- **the** $MaxColor$ **message**: it is a new message sent at the end of the coloring algorithm to inform the root of the tree of the number of colors used in the tree. This number defines the size of the active period in a cycle. The $Maxcolor$ message contains:
  - the node identifier of the sender,
  - its sequence number,
  - the maximum color used by all the descendants of the sender node.

  The $MaxColor$ message is sent from any node N to its parent until reaching the root. This unicast message is acknowledged. When it reaches the root, it contains the maximum number of colors used.

Moreover, each node maintains local information concerning:

- its neighborhood with the fields described in Section 4.3.3.2. Only the information concerning the size of its neighborhood up to three-hop is replaced by the number of its descendant.
- its parent in $\mathcal{T}$,
- for each child in $\mathcal{T}$,
  - the node identifier,
  - the maximum color used by this node and its descendants,
  - a sequence number.

5.2.3 Algorithm

We now give the three-hop coloring algorithm for a data gathering application. It consists of a procedure $Process(ColorMessage)$ and a main program.

Procedure 4 Process($Color$message)

if there is a change in the 1, 2 or 3-hop neighborhood or in the tree **then**

  update $\mathcal{N}^3(N)$, update $\mathcal{T}$, $parent(N)$ and $\mathcal{D}esc(N)$

end if

if $|Desc(N)|$ has not yet been computed and the 1,2 and 3-hop neighborhoods as well as $|Desc(M)|$ for each child M are known **then**

  compute $|Desc(N)|$ by adding the values of $1 + |Desc(M)|$ for each M, child of N.

end if

maintain the priority and color of any node in $\mathcal{N}^3(N)$

if N is the node with the highest priority among the uncolored nodes in $\mathcal{N}^3(N)$ **then**

  N selects the smallest color unused in $\mathcal{N}^3(N)$ strictly higher than $color(parent(N))$

end if

Algorithm 5 Three-hop coloring algorithm of a data gathering tree

Main

repeat

  repeat

    N broadcasts (1-hop) its $Color$ message containing:

    - a sequence number seq incremented at each change in the $Color$ message fields, except the sequence number fields

    - its identifier, its priority and color if already assigned

    - its list of one-hop neighbors with their identifier, their sequence number, their priority and their color

    - its list of two-hop neighbors with their identifier, their priority and color if already assigned.

    N waits for the $Color$ message of its 1-hop neighbors

    upon receipt of a $Color$ message,

      N Process($Color$message)

  until all one-hop neighbors have received the $Color$ message of N with number seq

until all nodes in $\mathcal{N}^3(N)$ are colored

5.2.4 Computation of the number of colors

We now evaluate the number of colors needed to color all network nodes. We first assume that the neighborhood up to 3-hop does not bring additional constraints to the one represented in the tree.

Property 1: If at each level p in the tree, each node has the same number of children $nbchild_p$ and there is no additional constraints to the one depicted in the tree, then the number of colors used is equal to:
$nbcolor = 1 + \sum_{p=0}^{depth-1} nbchild_p$.

If $\mathcal{N}^3(N)$ introduces additional constraints to those given in the tree, there exists a node N that has for 1-hop, 2-hop or 3-hop a node N' that is neither its parent, grandparent, brother, uncle, child, nephew, grandchild. In such a case, colors cannot be reused so easily, we then get:

Property 2: If at each level p in the tree, each node has the same number of children $nbchild_p$ and there are additional constraints to the one depicted in the tree, then the number of colors used is equal to:
$nbcolor = 1 + \sum_{p=0}^{depth-1} nbchild_p + \sum_{N} \delta_{N,N'}$, with $\delta_{N,N'} = 1$ if and only if N cannot take a color in $[color(parent(N)) + 1, cmax]$ with $cmax$ the highest color used by the nodes N' colored before N. Initially $cmax = color(parent(N))+1$ and $cmax = cmax+1$ each time a new color is used to color N.

We can get the formula giving the color of a node, depending on the color of its parent and the colors already used in its neighborhood up to three hops.

Property 3: The color of any node N is given by:
$color(N) = 1 + color(parent(N)) + \sum_{N' \ already \ colored \ \in \mathcal{N}^3(N)} \delta_{N,N'}$
with $\delta_{N,N'} = 1$ if and only if N cannot take a color in $[color(parent(N)) + 1, cmax]$ with $cmax$ the highest color used by the nodes $N' \in \mathcal{N}^3(N)$ colored before N. Initially $cmax = color(parent(N)) + 1$ and $cmax = cmax + 1$ each time a color in $[color(parent(N)) + 1, cmax]$ cannot be used to color N.

5.2.5 Comparison with another tree coloring algorithm

In this section, to justify the change of the priority (considering the number of descendants instead of the cardinal of the up tree-hop neighborhood), we compare our coloring algorithm with SERENA applied to tree topologies without modifying the priority. In another terms, we keep $|\mathcal{N}^3(N)|$ as the priority of nodes. To ensure that each node is scheduled before its parent, the coloring algorithm proceeds level by level (from the lowest level: the level of the tree rout to the highest one) and applying the rules RC'1 and RC2 instead of rules RC1 and RC2, where the rule

RC'1 is:

- **Rule RC'1:** any node N colors itself if and only if:
  - any node N' in $\mathcal{N}^3(N)$ having a level strictly lower than the level of N is already colored,
  - and any node N' in $\mathcal{N}^3(N)$ of the same level as N but $|\mathcal{N}^3(N')| > |\mathcal{N}^3(N)|$ is already colored,
  - and any node N' in $\mathcal{N}^3(N)$ of the same level as N but $|\mathcal{N}^3(N')| = |\mathcal{N}^3(N)|$ and a smaller identifier than N is already colored.
- **Rule RC2:** node N takes the smallest color available in $\mathcal{N}^3(N)$ strictly higher than the color used by its parent.

The simple example depicted in Figures 5.3 and 5.4 shows that the number of colors used by our algorithm (here 4 colors) is smaller than this used by this other algorithm (here 5 colors). The reason comes from the fact that our algorithm favors the nodes with a high number of descendants. As they are colored first, they can get smaller colors unlike in the other algorithm. The nodes with a small number of descendants are colored at the end, but they can reuse colors used by nodes that are not in their up to 3-hop neighborhood. We will see in Section 5.2.5.1 that the number of colors used by our algorithm is considerably smaller than the number of colors used by the other algorithm using rules RC'1 and RC2.

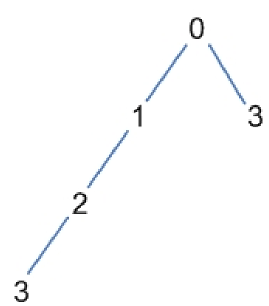

Figure 5.3: Tree colored with our algorithm.

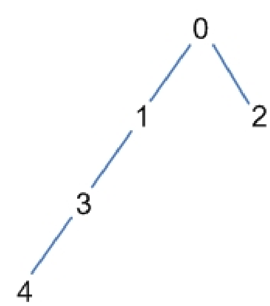

Figure 5.4: Tree colored with the other algorithm.

5.2.5.1 Comparative evaluation

We now compare the number of colors needed in our coloring algorithm with this needed by the other algorithm presented in Section 5.2.5 using $|\mathcal{N}^3(N)|$ as the priority and proceeding level by level. We consider a network density of 10.

Our algorithm exhibits very good performances as illustrated in Figure 5.5. This is due to the fact that it assigns a smaller color to nodes having a high number of descendants. Other nodes (with a small number of descendants) can reuse already used colors. Hence, the smallest number of colors is used. We can also observe that this trend increases with the number of nodes.

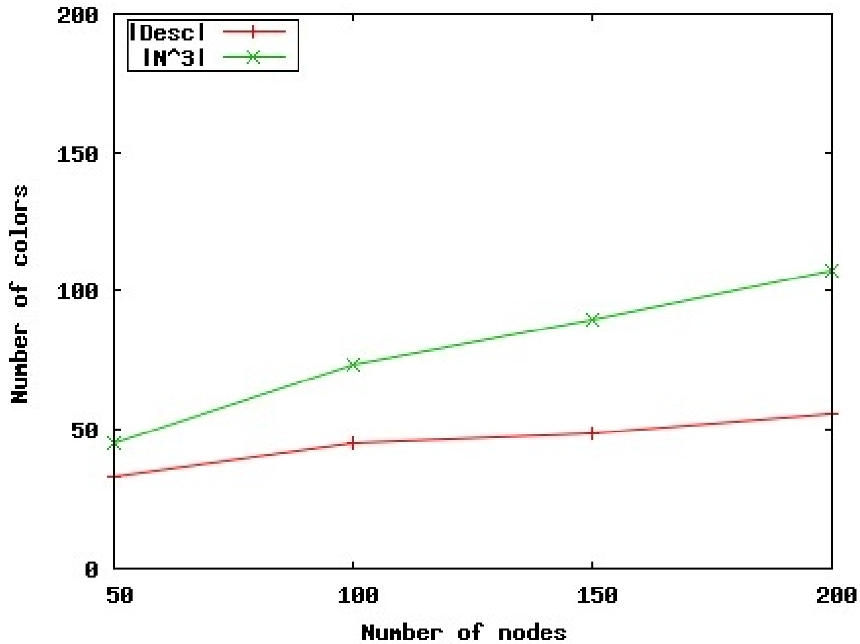

Figure 5.5: Comparative evaluation of the number of colors used by the $|Desc(N)|$ and $|\mathcal{N}^3(N)|$ algorithms.

In the next section, we will give a performance evaluation of our coloring algorithm adapted to tree topologies for data gathering applications. We compute different performance criteria such as the number of colors needed to color all network nodes, the average number of messages sent per node. We then show how this small number of colors benefits to the active period duration, the data gathering delay, as well as the network lifetime.

5.2.6 Performance evaluation

5.2.6.1 Simulation parameters

For performance evaluation purpose, we have considered different network configurations. Each configuration is characterized by a node number and a node density. The nodes are randomly deployed over the network area. The number of nodes ranges from 50 to 200, whereas the node density (i.e. the average number of one-hop neighbors per node) ranges from 10 to 20. For each configuration, we have run 10 simulations and the result depicted in the curves is the average of these 10 simulations. The data gathering tree is the tree of the shortest paths.

5.2.6.2 Number of colors

We now evaluate the number of colors needed by our algorithm to color all network nodes of the generated configurations. Simulation results are illustrated in Figures 5.6 and 5.7.

We first depict in Figure 5.6 the number of colors as a function of the number of nodes for different densities. We observe that this number strongly depends on node density and weakly on node number. As the color a node N cannot be reused in its neighborhood up to 3-hop, whose size is proportional to the density, this explains the strong dependency between the number of colors and the density. It is very interesting to notice that all curves depicting the number of colors are below the first bisectrix. This means that the number of colors is much smaller than the number of nodes, except for the network configuration of 50 nodes and a density of 20, where almost all nodes are 1, 2 or 3-hop neighbors. In other words, spatial reuse is always obtained for all the network configurations considered in the simulations. The higher the distance to the first bisectrix, the higher the spatial reuse. The average number of nodes using the same color is given by the number of nodes divided by the number of colors.

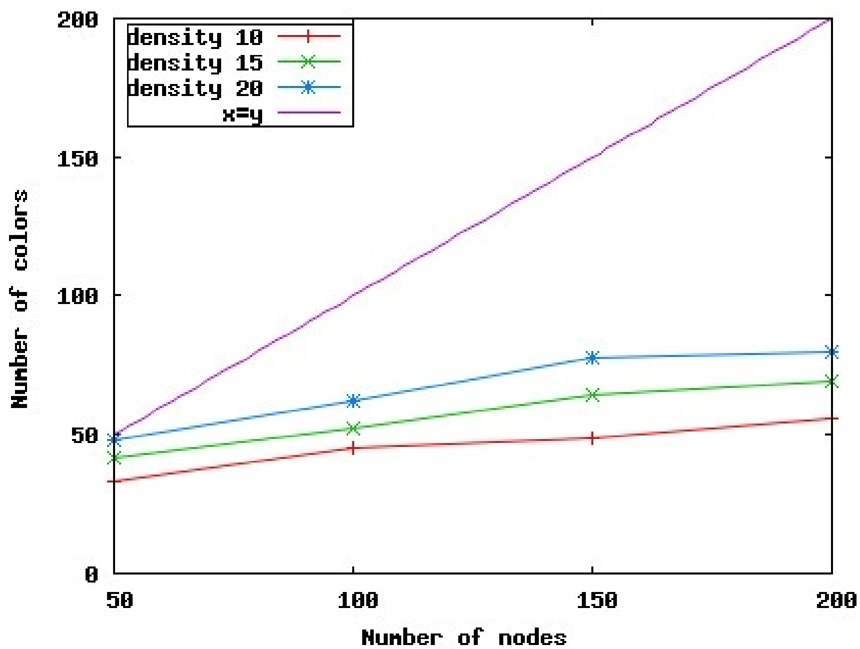

Figure 5.6: Number of colors as a function of the number of nodes and the density.

We now depict in Figure 5.7 the number of colors as a function of the tree depth for different densities. Notice that for a given density, the number of nodes increases with the tree depth. This is due to the fact that the color of a node is higher than the color of its parent. Moreover, the impact of tree depth seems to stabilize in the simulated configurations. The stabilization point is reached sooner for high densities.

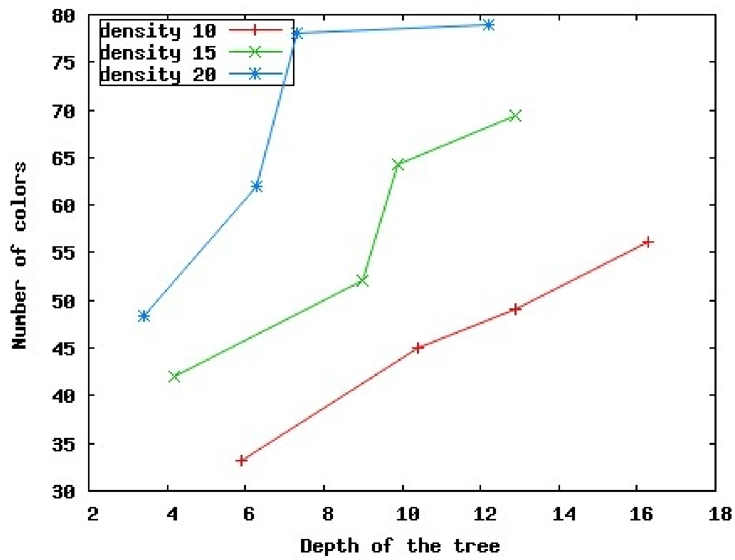

Figure 5.7: Number of colors as a function of the tree depth and the density.

5.2.6.3 Average number of messages sent by a node

In order to evaluate the overhead induced by the coloring algorithm, we compute the average number of messages sent per node in all the generated network configurations.

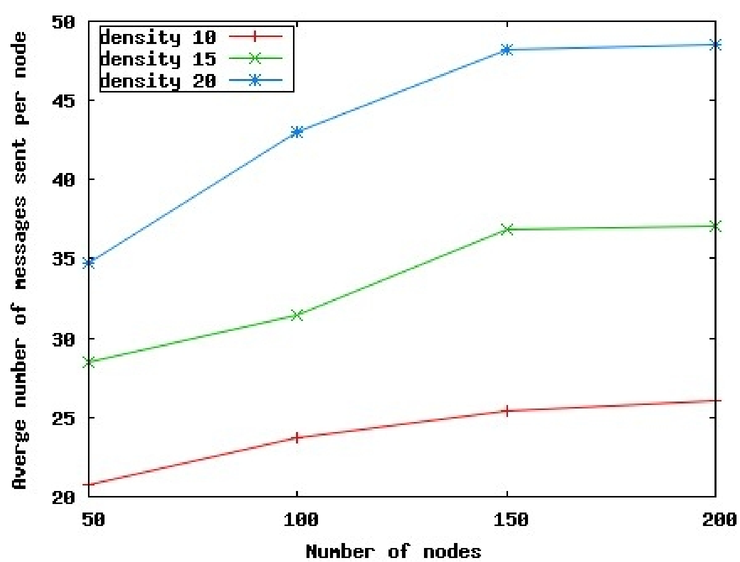

Figure 5.8: Average number of messages sent per node.

Simulation results are illustrated in Figure 5.8. The number of messages exchanged strongly increases with density and moderately with the number of nodes. It remains reasonable in all configurations.

5.2.6.4 Activity period duration and data gathering delay

The physical phenomenon that is monitored by the wireless sensor network determines the polling cycle of the sensors. This polling cycle comprises two periods, as represented in Figure 5.9: an activity period during which some sensors are active and an inactive period where all sensors are sleeping. Only the activity period is mandatory.

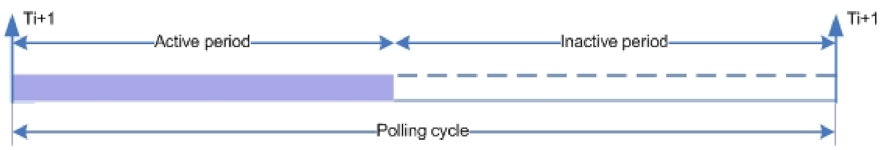

Figure 5.9: The active period in the polling cycle.

The coloring algorithm is used for slot assignment in the activity period. Slots are assigned to nodes per color instead per node like in classical TDMA: all nodes of the same color can transmit simultaneously without interfering. Hence, this space reuse leads to a considerable reduction of the activity period. The benefit is equal to $(numberofnodes - numberofcolors) \cdot slotsize$.

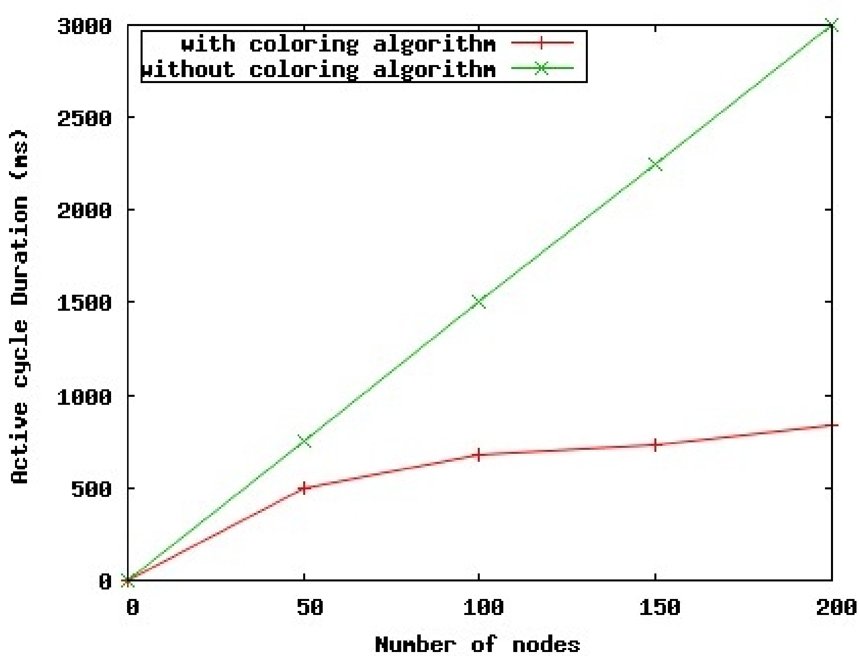

Figure 5.10: Duration of the active period as a function of the number of nodes.

Figure 5.10 shows the quantitative improvement brought by the coloring algorithm with regard to a classical TDMA. A slot size of 15 ms has been taken.

The data gathering delay is the time needed to collect at the sink all data transmitted by the sensors. If we assume that the slot has the capacity to contain the aggregated information transmitted by any child of the root (i.e.: the nodes having the highest quantity of information to transmit), then the time needed to collect the data from all sensors is equal to the activity period, whereas the latency is equal to the polling cycle duration.
Ensuring that each node has a color higher than its parent in the tree allows our coloring algorithm to reduce the data gathering delay and to obtain time-consistent data. In fact, in only one cycle, all data collected can reach the sink by assigning slots to colors in the decreasing order: this allows a child to transmit before its parent that can then aggregate the data received from its children. The increasing order is chosen in the reverse case of data dissemination. Moreover, collecting data in a single cycle guarantees time consistency of the data collected from the different sensors: for example temperature, humidity degree and pressure are measured in the same cycle.

5.2.7 Benefit brought by coloring

A coloring algorithm brings several advantages:

- an efficient use of the bandwidth by enabling spatial reuse and avoiding medium access contention.
- an increased network lifetime by enabling nodes to sleep during the activity period and the increased inactivity period. Optimizations can be brought in order to allow a node to sleep the soonest possible. For example, if a node has not detected any signal in the slot of its child during a predetermined duration, it can deduce that this child has nothing to transmit and goes back to the sleep state.
- a shorter delay to collect data from sensors. In a single cycle, all data can be collected, ensuring their time consistency.
- slots assigned according to the increasing order of colors reduces the time needed to disseminate data in the network: for example, information sent by the sink to all nodes.

5.2.8 Adaptivity of the coloring algorithm

5.2.8.1 Message loss

Our coloring algorithm tolerates message losses by means of the sequence number used to detect losses. A node N retransmits its *Color* message with sequence number *seq* as long as it has not received a *Color* message from all its one-hop neighbors containing the sequence number *seq* for N.

When a one-hop neighbor N' receives the *Color* message of N, it updates the neighborhood information locally stored. Two cases are possible:

- There is a change concerning:
  - either the color or the number of descendants of the node itself, or of a one-hop or two-hop neighbor,
  - or the appearance or disappearance of a one-hop or two-hop neighbor.

  N' increments its own sequence number seq' and sends a $Color$ message to its one-hop neighbors.

- Otherwise N' sends a $Color$ message without incrementing its sequence number.

It follows that any node will know a change in its neighborhood up to 3-hop, whatever the change is: node, color or number of descendants.

5.2.8.2 Tree change

In this section, we consider the case where the link between a node N and its parent in the tree is broken. This node N will first try to attach itself to another node N' in the tree. We assume that this link is an existing link: the parent of N is chosen among its existing one-hop neighbors. Hence this parent is already taken into account in $\mathcal{N}^3(N)$. We will see in the next section what happens when this assumption is not true.

Two cases are possible:

- the tree change does not impact the colors already assigned: no node in $\mathcal{N}^3(N)$ has the same color as N and $color(N) > color(N')$.
- the tree change creates a violation of the rule 2: the color of N is not higher than the color of its new parent N'. In this case, node N selects the smallest color available in $\mathcal{N}^3(N)$ belonging to $[color(N') + 1, min_{child} color(child) - 1]$. If there is no color available, N takes the smallest color of its children, and so on.

5.2.8.3 Topology changes

By topology change, we mean that a change in $\mathcal{N}^3(N)$ occurs: a new link is created or an existing link is broken. It can be caused by:

- node mobility: a node moves in the network area causing the breakage of its existing links and the creation of new ones.
- late node arrival: when a new node joins the already colored network, new links will appear.

In both cases, when a new link is created, it may have as consequence that two nodes that were not 1-hop, 2-hop or 3-hop neighbors become 1-hop, 2-hop or 3-hop neighbors. Hence, a color conflict between two nodes can occur. by definition (as presented in Chapter 4) *A color conflict occurs* *between two nodes having the same color when these nodes prevent each other or some neighbor destination to receive correctly the intended message because of a collision.*

Notice that in the absence of mobility and late node arrival, nodes that are 1, 2 or 3-hop neighbors never conflict by definition of the algorithm. Furthermore, this definition takes advantage of the capture effect and considers the only color conflicts where the intended destination is prevented to receive its message.

When a color conflict occurs between two nodes N and N', the node with the highest priority keeps its color whereas the node with the smallest priority takes another color according to rule RC2.

5.3 The optimized coloring algorithm for tree topologies

5.3.1 General principles

Three-hop coloring algorithm allows the spatial reuse when ensuring immediate acknowledgement to ensure a hop by hop reliability and to avoid collision. However, in the data gathering application with tree topology, communication is generally between a parent and its children in case of data dissemination or a node and its parent in case of data gathering. For these raisons, the coloring algorithm described above can be optimized. Instead of prohibiting a color to be used by two nodes that are one, two or three hops away, any node A and B do not share the same color only if:

1. Nodes A and B have the same parent.
2. Node B is one-hop neighbor of the parent of node A.
3. Node A is one-hop neighbor of the parent of node B.
4. Node A and B are one-hop neighbors.
5. The parent of node A and the parent of node B are one hop neighbors.

These five cases are illustrated by Figure 5.11, where a plain line represents a tree link, a dotted line represents a one-hop neighbor link that does not belong to the tree.

Data transmission is represented by a single arrow, whereas two arrows represent an acknowledgement transmission. Cases 1, 2 and 3 lead to a collision between the data sent by nodes A and B (at the parent of A in cases 1 and 2, at the parent of B in case 3). Cases 4 and 5 lead to a collision between data and acknowledgement. More precisely, with case 4, a collision occurs on node A between data sent by B and the acknowledgement sent by the parent of A. The symmetric case is also possible. With case 5, a collision occurs on the parent of A between data sent by A and the

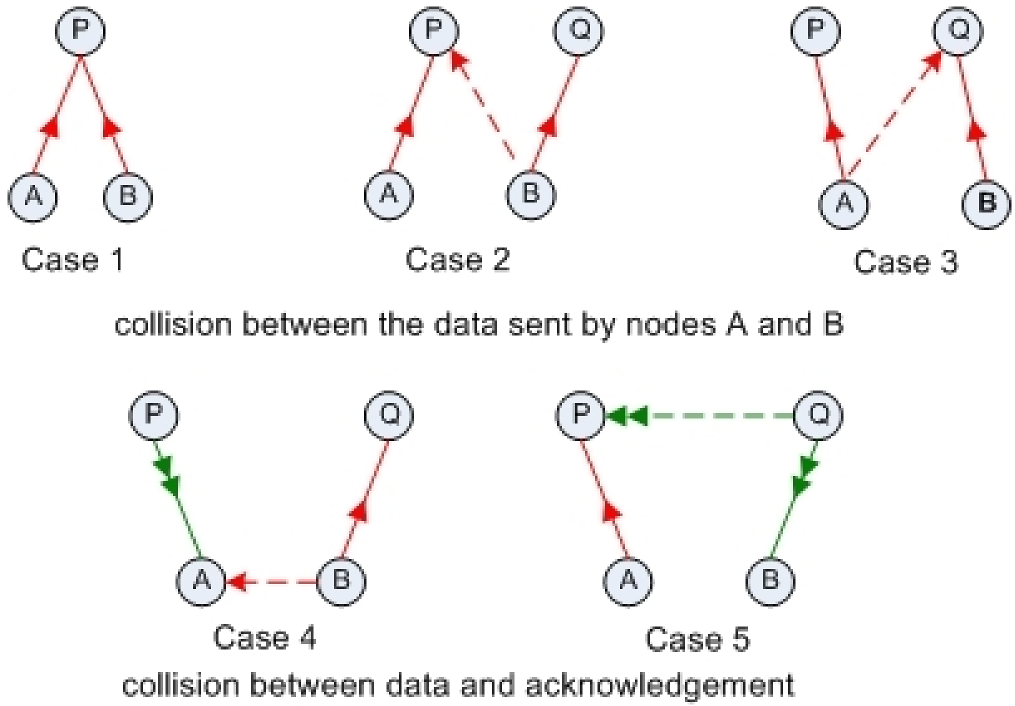

Figure 5.11: Different collision cases.

acknowledgement sent by the parent of B. The symmetric case is also possible.

In addition to these 5 collision cases, the coloring algorithm must ensure that each child has a color higher than its parent. The color of a node is used to schedule the medium access, but it also rules the activity periods of the nodes. A node must be awake during the slot attributed to its color and the slot attributed to its parent (respectively children) in a downstream (respectively upstream) communication. It can sleep the remaining time. Nodes having the same color can be active simultaneously.

We adopt the following notations:

- $\mathcal{B}(N)$: the brothers of N.
- $\mathcal{V}(N)$: the 1-hop neighbors of N.
- $\mathcal{VP}(N)$: the 1-hop neighbors of $parent(N)$.
- $\mathcal{CV}(N)$: the children of the 1-hop neighbors of N.
- $\mathcal{CVP}(N)$: the children of the 1-hop neighbors of $parent(N)$.
- $\mathcal{N}(N) = \mathcal{B}(N) \bigcup \mathcal{V}(N) \bigcup \mathcal{VP}(N) \bigcup \mathcal{CV}(N) \bigcup \mathcal{CVP}(N)$.

We use the same priority as the one described in Section 5.2.1 it is based on the number of descendants of a node in the tree. Hence, for any node N, we say that N has a higher priority than node $N' \in \mathcal{N}(N)$ if and only if:

- either N' meets $|Desc(N')| < |Desc(N)|$
- or $(|Desc(N')| = |Desc(N)|$ and $identifier(N') > identifier(N))$.

Our coloring algorithm is based on the two same rules RC1 and RC2 but with considering only the set of $\mathcal{N}(N)$ for each node and so rules RC1 and RC2 become:

- **Rule RC1:** any node N colors itself if and only if all nodes in $\mathcal{N}(N)$ having a higher priority than N are already colored.
- **Rule RC2:** node N takes the smallest color available in $\mathcal{N}(N)$ strictly higher than the color used by its parent.

The coloring algorithm is the same as the one described in Section 5.2.3, we should only replace $\mathcal{N}^3(N)$ with $\mathcal{N}(N)$. When the coloring algorithm ends, each node sends to its parent a $MaxColor$ message containing the highest color used by its descendants. Thus, the root knows N_{color} the number of colors needed to color all network nodes. The root can then schedule the medium access. Time slots are assigned to colors. The mapping can be very simple: one slot is assigned to each color. Nodes having the same color transmit simultaneously. If slots are assigned according to the decreasing order of colors and data aggregation is possible, then a single cycle is sufficient to collect data from all sensors. We notice that this coloring algorithm can also be used in case of data dissemination, the use of colors avoids collisions between nodes having received the information and being in charge of forwarding it. The medium access is then scheduled in the reverse order: according to the increasing order of colors. A single cycle is sufficient to disseminate information from the root to all network nodes. Hence, this algorithm allows an application to sometimes gather data and sometimes disseminate data.
In the following, we will compare our optimized coloring algorithm with classical TDMA and TDMA-ASAP a coloring algorithm proposed pour data gathering applications.

5.3.2 Comparison with TDMA-ASAP

5.3.2.1 TDMA-ASAP

TDMA-ASAP [96] is a node scheduling activity algorithm designed for tree topology in case of data gathering application. It is based on a coloring algorithm. Its goal is to find the smallest number of colors such that any node A and B do not share the same color only if:

1. Nodes A and B have the same parent.
2. Node B is one-hop neighbor of the parent of node A.

3. Node A is one-hop neighbor of the parent of node B.

Moreover, each child must be scheduled before its parent. To achieve that, a centralized algorithm is proposed which proceed level by level coloring beginning with the lowest level. Hence the root of the tree is the last node to be colored. The constraints taken in account by TDMA-ASAP correspond to our three first cases presented in Figure 5.11. Considering only these three cases and ignoring the case 4 and 5 means that immediate acknowledgment of unicast transmissions which is necessary to guaranteed the correct reception of messages is not considered. We will compare our proposition with TDMA-ASAP and evaluate the impact of considering the immediate acknowledgment in the spatial reuse.

5.3.2.2 Number of colors

We compare the number of colors with those used by TDMA-ASAP, where a color can be reused by several nodes, provided that cases 1, 2 and 3 are avoided and the color of a node is smaller than the color of its parent (the tree root being the last node to color). We have considered different network configurations characterized by a node number and a node density. The nodes are randomly deployed over the network area. The number of nodes ranges from 50 to 200, whereas the node density is fixed to 10. For each configuration, we have run 10 simulations and the result depicted in the curves is the average of these 10 simulations. The data gathering tree is the tree of the shortest paths. This study can also be extended to the case where the data gathering tree is built taking into account residual energy of nodes and their capacity to forward messages.

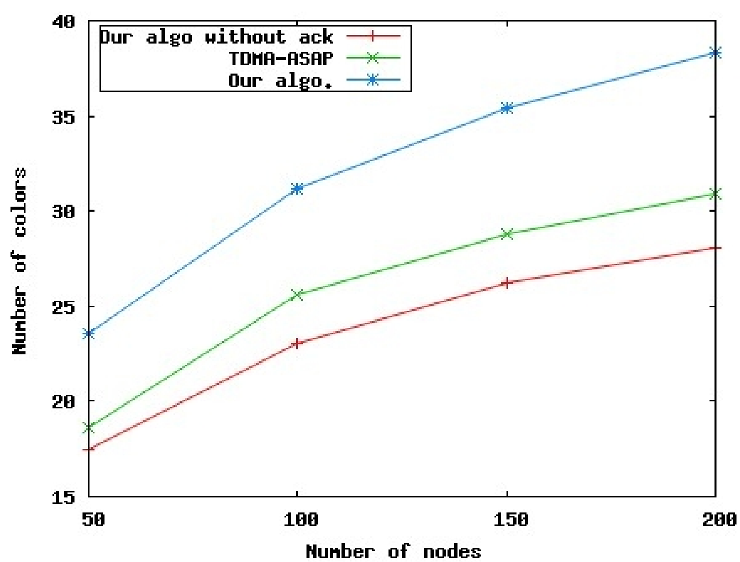

Figure 5.12: Number of colors as a function of the number of nodes.

Figure 5.12 depicts the number of colors as a function of the number of nodes for a fixed density of 10. it allows us to evaluate the price paid for the immediate acknowledgement. We observe an increase in the number of colors with regard to TDMA-ASAP. Without the immediate acknowledgement, our algorithm outperforms TDMA-ASAP, because of our optimization taking into account the number of descendants per node.

Our algorithm outperforms TDMA and the price paid for the immediate acknowledgement is reasonable as compared with TDMA-ASAP (e.g. 8 additional colors for 200 nodes).

5.3.2.3 Number of messages

Figure 5.13 shows that the average number of messages sent by a node to color itself is reasonable and does not depend on the number of nodes, unlike the maximum number of messages. Hence, our algorithm has a very satisfying complexity. A comparison with TDMA-ASAP is not given, because only a centralized version is presented in [96].

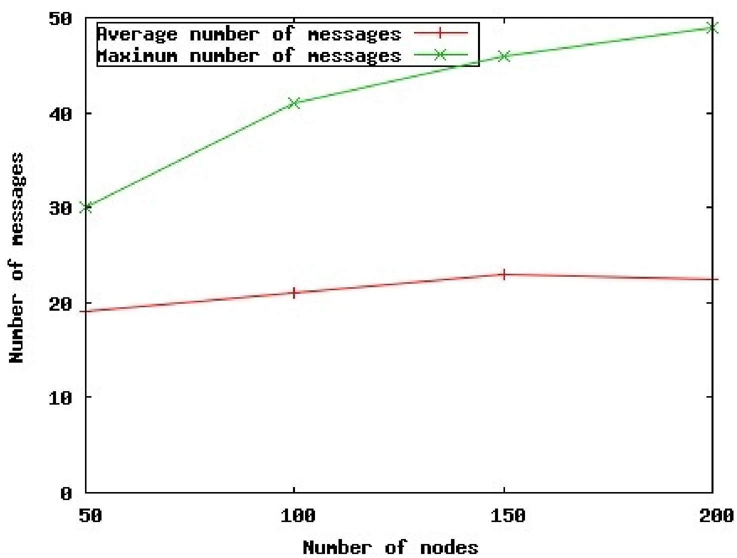

Figure 5.13: Number of messages as a function of the number of nodes.

5.4 Generalization

In this section we show how to generalize SERENA to have a generic solution that can be adapted to different user constraints. We recall that these constraints include:

- Types of communications:
  - Broadcast and unicast communications.
  - Only unicast communications.
  - Communication in the tree with the possible broadcast: the unicast communications are only between a node and its parent or between a node and its children in the tree. However, the broadcast communication is between a node and all its one-hop neighbors.
  - Communication in the tree only: a node can communicate only with its parent and its children in the tree.
- Immediate acknowledgement of unicast transmissions is required.
- Acknowledgement of unicast transmissions can be delayed or ignored.

From these constraints, we can define for any node N, on the one hand, the set of nodes $\mathcal{N}(N)$ that should not have the same color as N. The color message exchanged between one hop neighbors must contain all the fields needed to know the colors and priority of all nodes in $\mathcal{N}(N)$. On the other hand, the priority computation will depend on the application requirements. For instance, in general application the priority of N is equal to the cardinal of $\mathcal{N}(N)$. Whereas, in data gathering application, the priority of N is equal to the number of descendants of N. In the same way, the choice of the color depends on the application requirements. For example, in general application, the color chosen by N is the smallest color unused in $\mathcal{N}(N)$. Whereas in data gathering application, it is the smallest color unused in $\mathcal{N}(N)$ but higher than the color of its parent.

This generic solution tends to take into account the exact application requirements, neither more nor less than what is expected. This will contribute to make the overhead proportional to the exact application requirements. For example, in general applications, two-hop coloring is sufficient if no immediate acknowledgement of unicast transmissions is required. Otherwise, three-hop coloring is needed which uses more colors than two-hop coloring as expected. In data gathering application, if the broadcast is not used, the optimized algorithm can be used. It reduces the number of colors needed to color all the tree.

In the following we will show how to dimension the network resources for data gathering applications.

5.5 Network dimensioning with network calculus

5.5.1 Network calculus

Network calculus [98] is the tool to analyze flow control problems in networks with particular focus on the computation of bounds for the worst case. It has been successfully applied as a framework to derive deterministic guarantees on throughput, delay, and to ensure zero loss in packet-switched networks. Network calculus is based on min-plus algebra that applies to the deterministic analysis of queuing in networks. Network calculus theory assumes that for a given data flow:

- The input function $R(t)$ of an arrival process is the number of bits that arrive in the interval [0, t], with $R(0) = 0$, and R is wide-sense increasing function, i.e. $R(t_1) < R(t_2), \forall t_1 < t_2$.
- The output function of the flow, $R^*(t)$, is the number of bits that have left the system in the interval [0, t] with $R^*(0) = 0$, and R^* is wide-sense increasing function, i.e. $R^*(t_1) < R^*(t_2), \forall t_1 < t_2$.
- An arrival curve $\alpha(t)$ that upper bounds $R(t)$ such that $\forall s, 0 \leq s \leq t$, $R(t) - R(s) \leq \alpha(t-s)$. It is also said that R is α-smooth or R is constrained by α.
- A service curve $\beta(t)$. Consider a system S and a flow through S with R and R^*. S offers a service curve β to the flow means that the output flow during any given busy period [t, t+Δ] is at least equal to $\beta(\Delta)$.

What makes network calculus different from traditional queueing theory is that it is concerned with worst case rather than average case or equilibrium behavior. In fact, the knowledge of the arrival and service curves enables us to determine performance bounds, namely the delay bound D_{max} given by the maximum horizontal distance between $\alpha(t)$ and $\beta(t)$, which represents the worst-case delay of the message traversing system S, and the backlog bound Q_{max} given by the maximum vertical distance between $\alpha(t)$ and $\beta(t)$, which represents the minimum buffer size requires inside the system S (see Figure 5.14.

D_{max} is expressed as follows: $D_{max} = sup_{t\geq 0}\{inf(s \geq 0/\alpha(t) \leq \beta(t+s))\}$
Q_{max} is expressed as follows: $Q_{max} = sup_{t\geq 0}(\alpha(t) - \beta(t))$

Example of max delay and max backlog bound computation. We suppose a linear arrival curve $\alpha(t) = b + rt$ that receive a service curve $\beta_{R,T}(t) = R(t-T)^+$, where $R \geq r$ is the guaranteed bandwidth, T is the maximum latency of the service and $(x)^+ = max(0, x)$. This example is illustrated in Figure 5.14.

The delay bound D_{max} given by the maximum horizontal distance between $\alpha(t)$ and $\beta(t)$ is computed as follows:

$$\beta_{R,T}(D_{max}) = \alpha(0) \Leftrightarrow R(D_{max} - T) = b \Leftrightarrow D_{max} = \frac{b}{R} + T$$

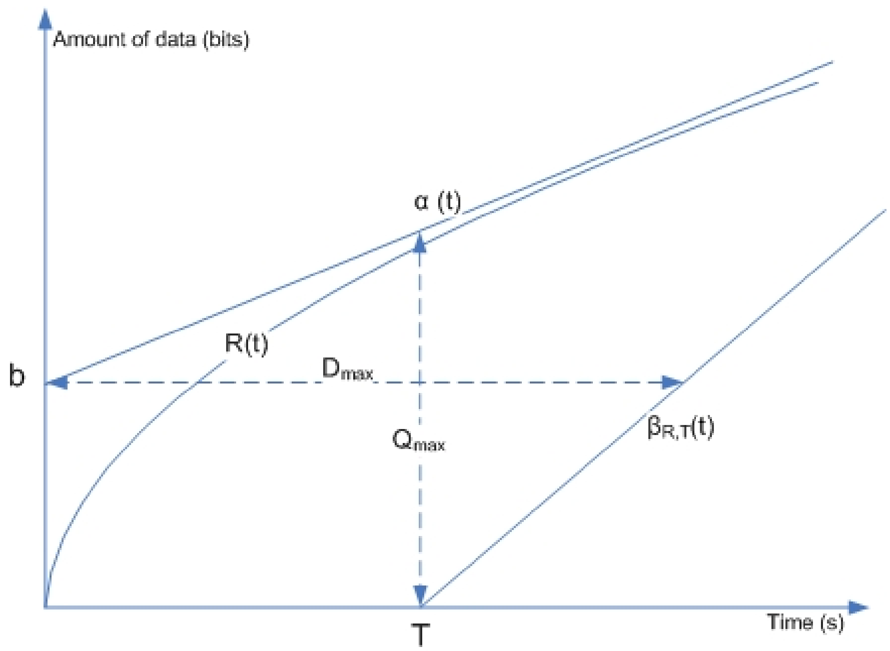

Figure 5.14: Delay and backlog bounds.

The backlog bound Q_{max} given by the maximum vertical distance between $\alpha(t)$ and $\beta(t)$ is computed as follows:
$Q_{max} = \alpha(T) \Leftrightarrow Q_{max} = b + rT$

In our analysis we will use the previous linear arrival curve and the rate latency service curve.

5.5.2 Comparison of our algorithm with ZigBee

In this section, we compare the performances obtained by our algorithm with those of a classical ZigBee cluster tree published in [102]. We adopt their framework and compute the results for our algorithm in order to evaluate its benefits.

5.5.2.1 ZigBee presentation

We first recall some ZigBee concepts. We distinguish two types of sensors as in ZigBee: (1) sensors with full functionality (FFD in ZigBee) that are able to route messages; these nodes are called routers in the following and (2) sensors with reduced functionality (RFD in ZigBee) that are unable to route messages and must be attached to a sensor with full functionality. The structure of a ZigBee superframe is depicted in Figure 5.15.

The duration between two successive beacons is called Beacon Interval, denoted BI. It consists of an active period, called Superframe Duration, denoted SD and a potential inactive period. SD

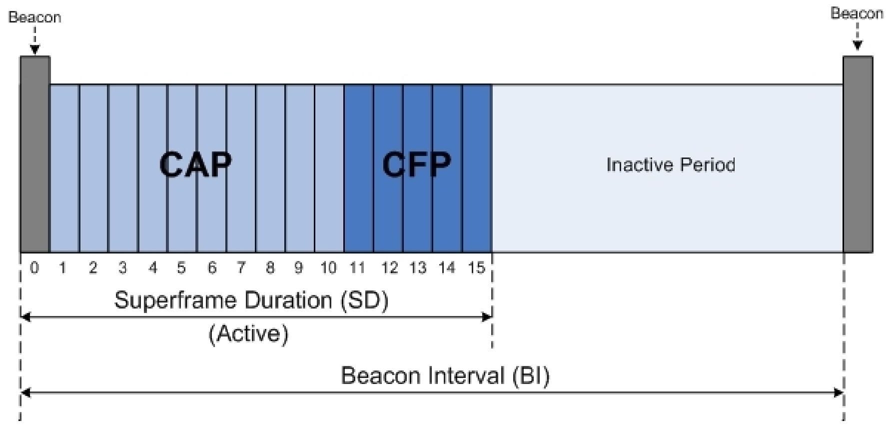

Figure 5.15: ZigBee superframe structure.

comprises a contention access period, denoted CAP and a contention free period, denoted CFP. The beacon interval and the active period duration are expressed by means of the beacon order, denoted BO and the superframe order, denoted SO. We have $BI = aBaseSuperframeDuration \cdot 2^{BO}$ symbols and $SD = aBaseSuperframeDuration \cdot 2^{SO}$ symbols with $0 \leq BO \leq SO \leq 14$.

5.5.2.2 Framework

For a sake of simplicity, we adopt the notations defined in [102], where the authors use network calculus to compute the buffer dimensioning, the bandwidth reserved at each router and the maximum end-to-end delay for a classical ZigBee cluster. A cluster tree is defined by three variables:

- N_{router}: the maximum number of children routers (i.e. FFD children) per router,
- N_{RFD}: the maximum number of children nodes that are not routers (i.e. RFD children) per router,
- $maxDepth$: the depth of the router tree, it is also the maximum distance (expressed in hop number) of a router to the root of the tree considered.

We assume as in [102] that the data arrival at each node (router or child node) is bounded at time t by the arrival curve $\alpha_{data}(t) = b_{data} + r_{data}.t$, where b_{data} is the maximum burst size and r_{data} its average rate. As for the service curve, we assume that each node has a service guaranteed from its parent router corresponding to the service curve $\beta_{data} = R_{data}.(t - T_{data})$, where R_{data} is the guaranteed bandwidth and T_{data} the maximum service latency.

According to [102], we can deduce the input rate $r^{input}_{maxDepth-i}$ and the output rate $r^{output}_{maxDepth-i}$ of a router at depth $maxDepth - i$:

$$r^{input}_{maxDepth-i} = r^{output}_{maxDepth-i} = \left(\sum_{j=0}^{i} N^j_{router}\right).r^{input}_{maxDepth}.$$

As in [102], we assume that:

- Each SD is divided into 16 time slots. Moreover, $SO = 0$, hence $SD = 15.36$ ms.
- The maximum CFP (Contention Free Period) length is equal to $L_{CFP} = 14$, only two time slots are allocated to CAP (Contention Access Period). This supposition is just used to illustrate our purpose. Notice that in the IEEE 802.15.4 standard the minimum CAP length is equal to 7.08 ms which correspond to 8 time slots.
- Each child node receives at most one time slot from its parent. Hence, the number of time slots reserved to N_{router} children is equal to $(L_{CFP} - N_{RFD})$.

5.5.2.3 Cycle dimensioning

With regard to cycle dimensioning, the two following feasibility conditions must be met by the colored tree:

- FC1: Each router when it is allowed to access the medium guarantees medium access to all its children nodes/routers in the same *cycle*.
- FC2: All routers are guaranteed to access the medium each *cycle*.

These feasibility conditions become for a ZigBee cluster tree:

- FC1: $N_{router} + N_{RFD} \leq 7$, according to the ZigBee standard.
- FC2: We distinguish two cases depending on the SD size of routers.

1. *First case: all routers have the same SD size:*
   With colors, we get: $BI \geq N_{color} \cdot SD$ which leads to the minimum BO
   $BO_{minc} = \lceil log_2(N_{color})\rceil$.
   Without colors, and considering that routers activity do not overlap, we would have get:
   $BI \geq SD \cdot \sum_{j=0}^{maxDepth} N^j_{router}$ as in [102], leading to
   $BO_{min} = \lceil log_2(\sum_{j=0}^{maxDepth} (N^j_{router})\rceil$.
   As a consequence, colors allow the beacon period to be smaller of a factor BO_{minc}/BO_{min}. In a dual way, for the same beacon period, a higher number of routers can be supported.

In [103], it has been shown that in a ZigBee star, the bandwidth guaranteed by one allocated time slot in a superframe of minimum size (i.e. $SO = 0$), denoted R^*_{TS}, is equal to 9.38kbps. In a ZigBee cluster tree with a superframe of minimum size, the bandwidth guaranteed by one allocated time slot is equal to $R_{TS} = \frac{R^*_{TS}}{2^{BO_{minc}}}$ in a colored tree instead of $\frac{R^*_{TS}}{2^{BO_{min}}}$. For instance, for the example illustrated in Figure 5.16 with 15 routers, assuming that any two RFDs with different parents do not interfere, we obtain 7 colors, $BO_{minc} = 3$, $R_{TS} = 1.1725$ kbps, instead of $BO_{min} = 4$, $R_{TS} = 0.586$ kbps with [102].

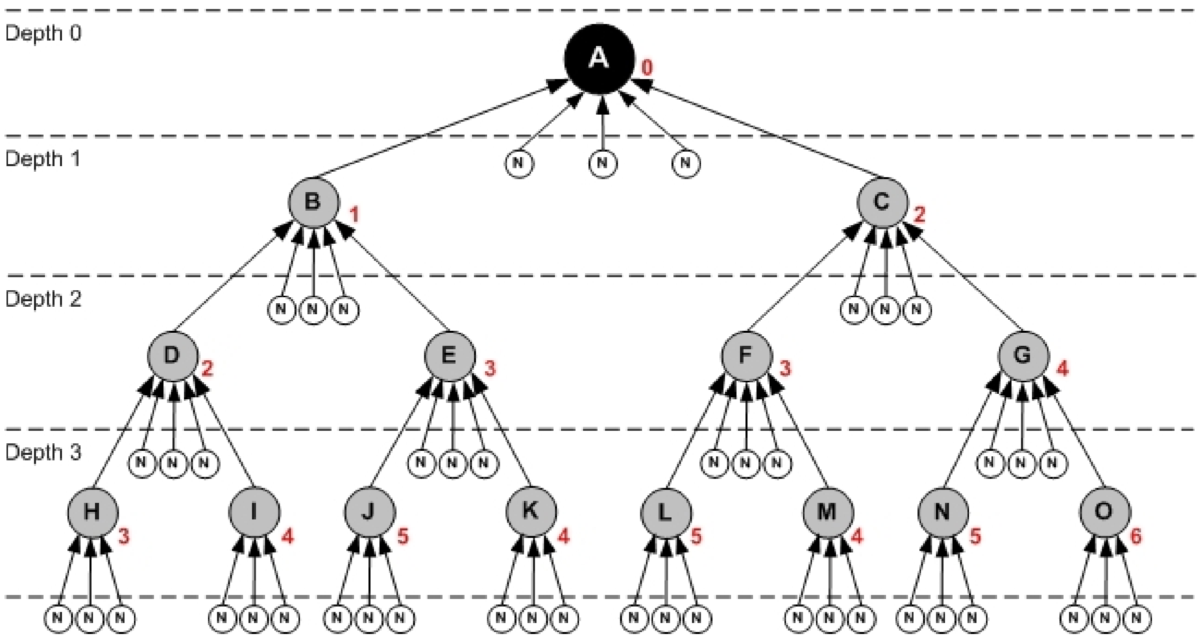

Figure 5.16: A colored tree.

2. *Second case: SD size depends on router depth:*
   With colors, we get:
   $$BI \geq N_{color} \cdot (CAP_{min} + N_{RFD} \cdot T_S)$$
   $$+N_{router} \cdot T_S \cdot \sum_{c=0}^{N_{color}-1} max(N_{j+1})_{j=0..maxDepth-1},$$
   where c is a color used at depth j and T_S the slot size.
   Without colors, we would have obtained:
   $$BI \geq N_{router} \cdot (CAP_{min} + N_{RFD} \cdot T_S)$$
   $$+N_{router} \cdot T_S \cdot \sum_{j=0}^{maxDepth} N^j_{router} \cdot N_{j+1}.$$
   Here also, colors allow either a smaller beacon period or a higher number of routers.

5.5.2.4 Maximum throughput

With regard to the total arrival rate at a router, the following feasibility condition must be met:
• FC3: The total arrival rate at the router must be limited in order not to exceed the guaranteed bandwidth that the router can provide.
Let R_{TS} be the bandwidth guaranteed by one allocated time slot. According to FC3, the maximum input rate from a child router at a depth 1 (i.e. r_1^{input}) is equal to:
$r_1^{input} = \lfloor \frac{L_{CFP}-N_{RFD}}{N_{router}} \rfloor . R_{TS} = (\sum_{j=0}^{2} N_{router}^{j}) . r_{maxDepth}^{input}$.
Hence, we can deduce the maximal input data rate [102]:
$r_{data}^{max} = \lfloor \frac{L_{CFP}-N_{RFD}}{N_{router}} \rfloor . \frac{R_{TS}}{(\sum_{j=0}^{2} N_{router}^{j})\,(N_{RFD}+1)}$.

For $L_{CFP} = 14$, we get $r_{data}^{max} = 0.21$kbps with our algorithm. However, in a classical ZigBee cluster tree, r_{data}^{max} is equal to $r_{data}^{max} = 0.104$ kbps. **Since our algorithm introduces parallelism in the tree, the network supports a higher input traffic rate.**
For the comparison with the ZigBee cluster tree [102], we assume $b_{data} = 200$ bits and $r_{data} = 0,1$ kbps.

Time slots reserved to each router. We can now compute the number of time slots that each router can obtain from its parent. This number indicates the quantity of bandwidth that is allocated to this router to send its data. This bandwidth must satisfy FC3. Hence, the minimum number of time slots required by each router at depth $maxDepth - (i+1)$, with $i \in [0, maxDepth - 1]$ is equal to: $N_{maxDepth-(i+1)}^{TS} = \lceil \frac{r_{maxDepth-i}^{input}}{R_{TS}} \rceil$.

Bandwidth reservation at each router. The bandwidth reserved in each router is proportional to R_{TS} the bandwidth guaranteed by one allocated time slot :
$R_{maxDepth-i} = N_{maxDepth-(i+1)}^{TS} . R_{TS}$, where $R_{maxDepth-i}$ is the bandwidth reserved by each router at depth $maxDepth - i$.
In Figure 5.17, we represent for each depth of the tree, the bandwidth required and the bandwidth reserved at each router, with our algorithm and ZigBee. At each depth, the requested bandwidth is smaller than the reserved bandwidth, which is higher with our algorithm than with pure ZigBee.

5.5.2.5 Buffer dimensioning at each router

With regard to buffer dimensioning, the following feasibility condition must be met:
• FC4: The buffers at each router are dimensioned in such a way that no data is lost.
According to [102], the buffer size required in a router at depth $maxDepth - i$, denoted $Q_{maxDepth-i}$ meets:

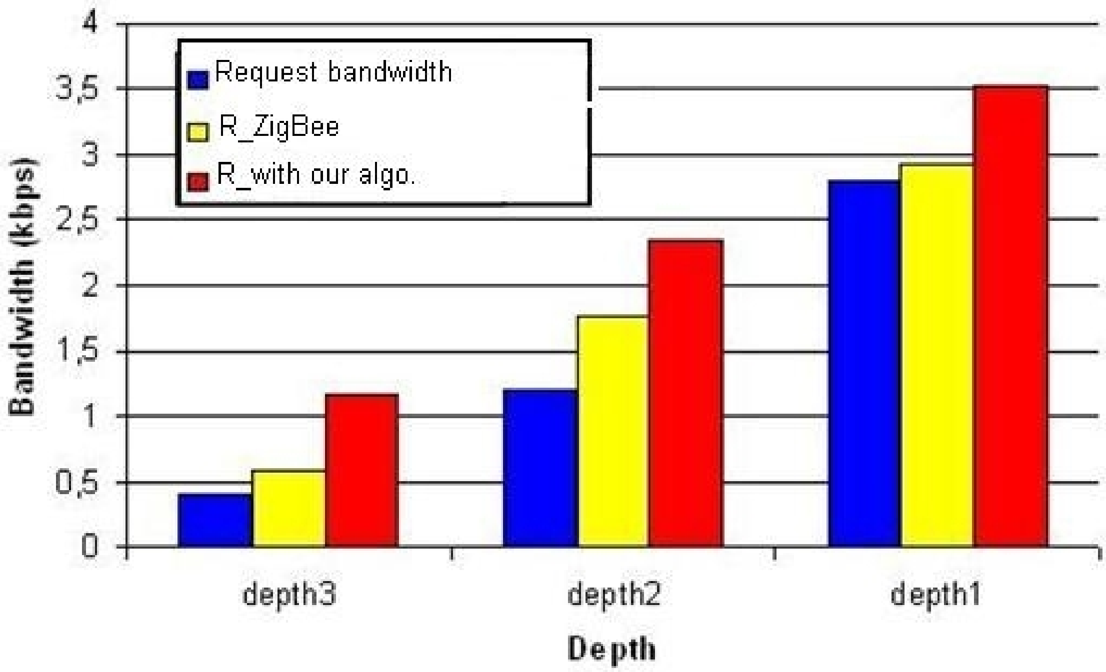

Figure 5.17: Bandwidth reserved with and without our algorithm.

$$Q_{maxDepth-i} = (\sum_{j=0}^{i} N_{router}^{j}).b_{maxDepth}^{input} + \sum_{j=0}^{i}\left(N_{router}^{i-j}.(\sum_{k=0}^{j} N_{router}^{k}).r_{maxDepth}^{input}.T_{maxDepth-(j+1)}\right).$$

where $b_{maxDepth}^{input}$ denotes the burst of the input at a router of depth $maxDepth$.
This equation is used to compute and compare results obtained by our algorithm with those obtained by ZigBee cluster tree. In Figure 5.18, we depict the buffers required at any router as a function of its depth. It is clear that Computation results show that our solution outperforms ZigBee scheduling approach that needs higher buffer sizes. The obtained gain increases when the depth of routers decreases and reaches 14% at depth 1 for the example illustrated in Figure 5.16. Moreover, this benefit increases with the size of the tree considered.

5.5.2.6 Maximum delay computation

One aim of our solution is to improve the end-to-end delays in ZigBee cluster tree networks. In this section, we will compare the end-to-end delay in our algorithm with the end-to-end delay in ZigBee cluster tree. First, we compute the maximum delay to cross a level as follows [102]:

$$D_{maxDepth-i} = \frac{b_{maxDepth-i}^{input}}{R_{maxDepth-(i+1)}} + T_{maxDepth-(i+1)}.$$

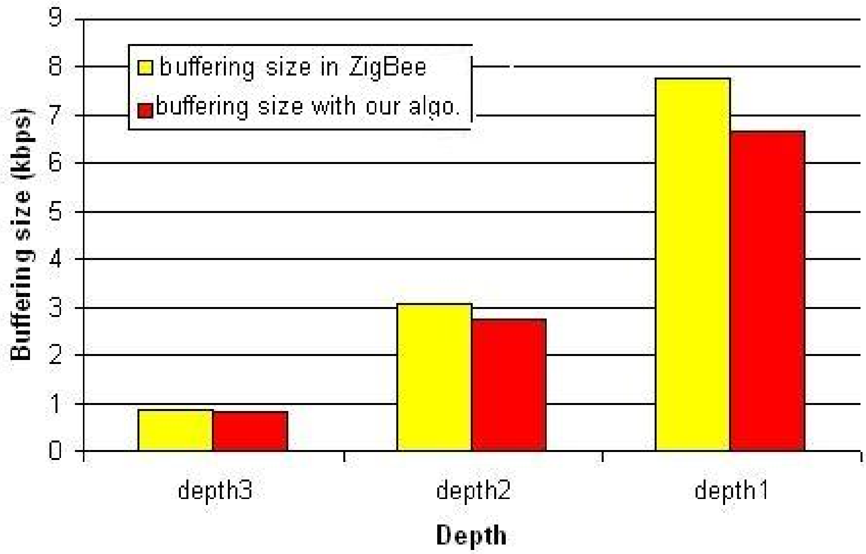

Figure 5.18: Buffer dimensioning at each router with and without our algorithm.

where $D_{maxDepth-i}$ and $b^{input}_{maxDepth-i}$ are respectively the maximum per hop delay and the input burst of a router at depth $maxDepth - i$, whereas $R_{maxDepth-(i+1)}$ and $T_{maxDepth-(i+1)}$ are the guaranteed bandwidth and the maximum service latency at depth $maxDepth - (i + 1)$.

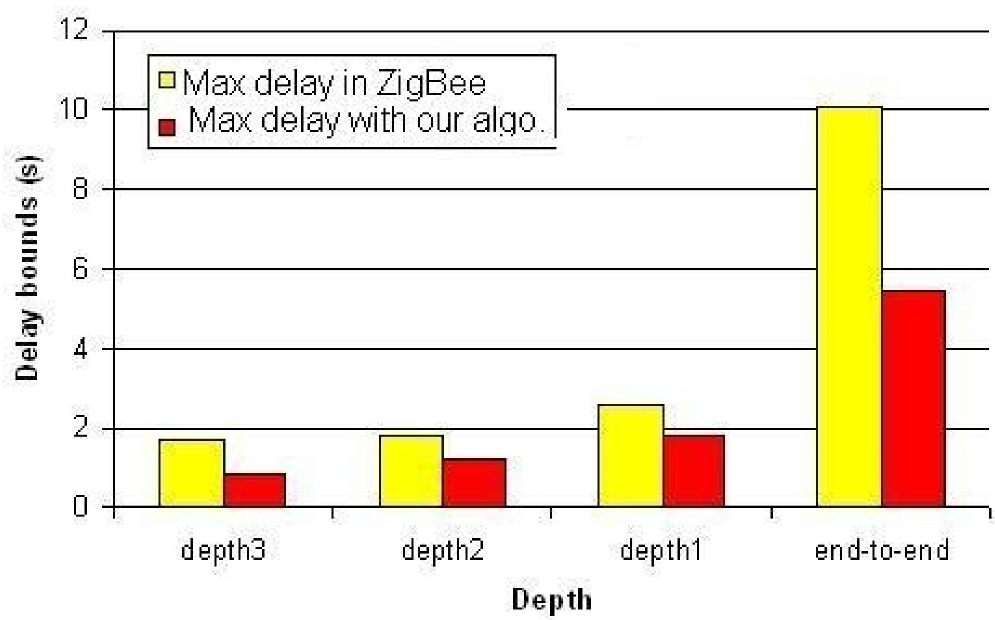

Figure 5.19: Maximum delay per hop and maximum end-to-end delay.

The maximum end-to-end delay is computed as described in [104]. Figure 5.19 depicts the maximum delays obtained by our algorithm and the ZigBee cluster tree. More precisely, we compare the per hop delay and the maximum end-to-end delay. Results show that at each depth the maximum per hop delay is smaller in our algorithm than in ZigBee cluster tree. That is due to the parallelism introduced by our algorithm. Indeed, the use of colors allows several routers to transmit simultaneously. Hence, our algorithm allows us to maximize the bandwidth allocated to each router as well as to reduce the latency and the service time. Consequently, the maximum end-to-end delay is improved as shown in Figure 5.19. We notice that the maximum end-to-end delay in our algorithm is almost a half of the end-to-end delay in ZigBee cluster tree.

5.6 Conclusion

Improving throughput and decreasing end-to-end delays while saving node energy is a crucial problem for data gathering applications in wireless sensor networks. We have proposed a solution based on graph coloring ensuring that data gathering or data dissemination takes only a single cycle, assuming data aggregation. Nodes with the same color can transmit simultaneously, achieving spatial reuse of the bandwidth. The benefits brought by this coloring algorithm are quadruple:

- the bandwidth is used more efficiently, taking advantage of spatial reuse and avoiding medium access contention by means of colors;
- the nodes spare their residual energy in sleeping while they have no message to send or to receive. In an upstream (respectively downstream) communication, a node is awake during the slots allocated to its children (respectively its parent) and its slot; it sleeps the remaining time, saving energy.
- the data transmitted by the sensors can be collected in a single activity period, ensuring their time consistency.
- colors can also be used to disseminate information from the sink to all sensor nodes in a single cycle.

We first began with adapting the SERENA's three-hop coloring algorithm to tree topologies. We presented the benefit brought by this new algorithm for data gathering applications. Then, we have optimized this algorithm. This optimization significantly improves the performance of the coloring algorithm designed for tree topologies. It is useful when all nodes in the network are routers. However, in a network like ZigBee where there are two types of nodes: RFD and FFD (router) and only routers execute the coloring algorithm, RFD nodes take the same color as their parents. Nevertheless, these RFD can interfere with other nodes having the same color but not the same parents. Then, collisions can occur. To avoid this problem, we propose to use the SERENA'three-hop coloring algorithm adapted to tree topology if in the network there nodes not able to route data.

We have compared the performances of our coloring algorithm optimization with these obtained in [102] for a ZigBee tree, highlighting high benefits. This solution also ensures time consistency of collected data. The benefits brought by our solution increase with the size of the tree. With regard to other coloring algorithms like TDMA-ASAP, our algorithm uses a lightly higher number of colors due to the immediate acknowledgement of unicast transmissions, which is needed for a reliable delivery. Without this constraint, it would use a smaller number of colors.

Chapter 6

Cross layering and integration of energy efficient techniques

6.1 Introduction

The characteristics of the layered architectures are not sufficiently flexible to cope with the specificities of wireless ad hoc and sensor networks. These specificities include:

- dynamically changing network conditions due to mobility or versatility of the propagation conditions on the physical medium,
- network resources of limited capacity: limited bandwidth, limited amount of energy for battery operated nodes,
- radio interferences...

They bring new constraints in designing protocols for wireless sensor networks. These protocols must be energy efficient and dynamically adaptive. Moreover, they must satisfy the quality of service desired by the applications. To meet these requirements, there were two classical approaches to design new protocols:

- Either the generic one that design protocols to suit a very high number of applications. The drawback is that it can lead to poor performances for some applications or can be to much expensive in terms of network resources. We can cite for instance [61] for scheduling node activity in sensor networks.
- Or, at the opposite, the specific approach consists in designing protocols optimized for a given type of application (e.g. data gathering by a sink node). This results in very good performances, however it provides no flexibility. As soon as the real application differs a little bit from the one considered, it can no longer be applied. Hence, this approach is not largely adopted.

Now an hybrid approach is possible, keeping the generality of the first approach and some optimizations of the second one. This third approach is based on cross layering. It optimizes the performance of communication, taking advantage of some knowledge about the application and its environment.

In this chapter, in the first step, we will focus on cross layering and detail the benefits obtained by this approach to improve the performances of EOLSR and SERENA. In Section 6.2, a brief overview of the state of the art related to cross layering is given. In Section 6.3, we focus on cross layering used by the routing protocol. we first consider the crosslayering between the application layer and the routing layer. We then consider cross layering between the routing layer and the MAC layer and evaluate the benefits on the performance of the routing protocol. In Section 6.4, we address the cross layering between the SERENA and both the application layer and the MAC layer. In the next step, we will present our contribution in the OCARI project. We will first describe this project in Section 6.6: its goal, architecture and different components. Then we will present our contribution in this project which consists in designing the energy efficient network layer. Finally, we conclude in Section 6.7.

6.2 State of the art

Cross layer design breaks away from traditional network design, where each layer of the protocol stack operates independently and exchanges information with adjacent layers only. In the cross layer approach, information is exchanged between non-adjacent layers of the protocol stack. We can distinguish three architectures for the cross layer:

- The first architecture is a non layered architecture. This architecture is very flexible and adapts to any type of application. However, the adaptation to a given application is very complex. This architecture has been proposed in [77]. However its inherent complexity makes it difficult to maintain and evolve when new requirements appear.
- In the second architecture, non adjacent layers can communicate with each other to ensure cross layering optimization. This architecture depends on the requirements of a given application. For instance, video delivery in wireless networks has been considered in [75]. It is difficult to adapt this architecture to different types of application. Such an architecture has been used also in [76]. It is illustrated in Figure 6.1.
- The idea of the third architecture is to keep the functionality of the protocol stack allowing non adjacent layers to communicate via an intermediate entity. The advantage of this architecture relies on its genericity that adapts to different types of applications. This architecture has been proposed in [78, 79] and is illustrated in Figure 6.2.

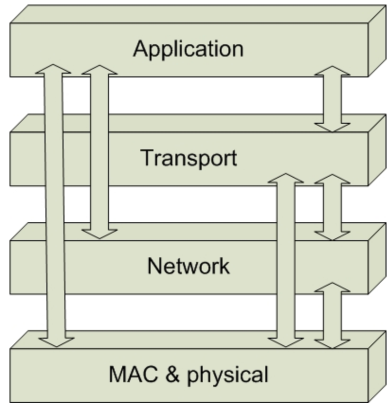

Figure 6.1: Interaction between adjacent and non adjacent layers

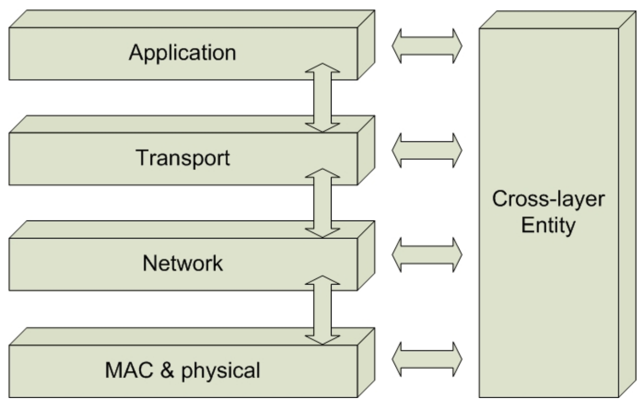

Figure 6.2: Cross-layering architecture with cross-layer entity

In our work, we adopt this third solution for the cross-layering architecture, because of its higher genericity. We show now how to improve the network performance using such cross layering.

6.3 Cross layering with EOLSR

6.3.1 Routing and application cross layering

The first idea to optimize EOLSR, taking advantage of the knowledge provided by the application, would be to reduce the overhead induced by $Hello$ and TC messages. The second idea consists in maintaining only routes to strategic nodes identified by the application. The third idea would be to select more stable routes, knowing the mobile nodes and possibly their trajectory. We will now detail these ideas.

6.3.1.1 Reducing the induced overhead

To reduce the overhead induced by the *Hello* and *TC* messages, we can identify at least three ways to proceed:

- The simplest way consists in increasing the *Hello* period and the *TC* period. However, *Hello* messages are used to discover link breakages and link appearances; a higher period would result in an increase of protocol latency and a poor reactivity. A trade-off must be found. Some knowledge of the application can be very useful in selecting the good value. For instance, the application can run different phases. It would be judicious to adapt the periods to each application phase.
- The second way is based on the following constatation: *Hellos* are sent only one-hop away, whereas *TCs* are sent all over the network. The use of MPRs allows to optimize broadcasts in reducing the number of *TC* retransmissions and usually exhibits better performance than other techniques [81]. In large networks, the Fish Eye option [44] can be used to tune the refreshment period of an information as a function of the distance: a node close to the origin of the information is refreshed more frequently and consequently has the most up-to date version. Applied to the *TCs*, it means that a MPR node sends its TC once per period in its close area, and for instance, once per two periods in the medium area, and once per four periods all over the network. This results in a decreased bandwidth consumption as shown in [105], where the benefit can reach up to 50% in networks of 1000 nodes with densities of 50 nodes.
- The third way consists in maintaining only useful routes. These useful routes are known either by their source or their destination (e.g. sink node in data gathering applications). Hence, only sink nodes send TC messages. These TCs allows all nodes in the network to build a route to reach the sink node originator of TC. This way to optimize the routing protocol is described in more detail in Section 3.5 in Chapter 3.

6.3.1.2 Other improvements

There are also other improvements of the routing protocol made possible by the knowledge from the application. For instance, the application can know the movement of nodes. In some applications, a node moves according to a well known trajectory. In a VANET for instance, a node embedded in a car moves on the highway. The knowledge of this information will help the routing protocol to decide on the links to choose. Hence, the EOLSR protocol will select the MPRs of the moving node N among the nodes with the most stable links.

In the same way, the application knows which nodes are mobile. This knowledge can be used by the routing protocol. These mobile nodes will then refresh their *Hellos* more frequently than static nodes. A shorter link refreshment period leads to an earlier detection of broken links and new links. The $Hello\_Interval$ can be set per link and adapted to node mobility. The overhead, proportional to the refreshment period, can be calibrated by node velocity.

6.3.2 Routing and MAC cross layering

There are several possibilities to improve the routing performance by use of MAC information. We can cite, for instance:

- The knowledge of the power of the received signal enables to select more stable routes and improves routing protocol reactivity, as we will see in Subsection 6.3.2.1;
- The success of the attachment procedure between two nodes enables to conclude that these nodes are neighbors, without waiting for the exchange of at most 3 Hellos, as we will see in Subsection 6.3.2.2;
- The knowledge of the bandwidth available at the considered node enables to select a route providing the requested bandwidth, if it exists.

6.3.2.1 Knowledge of the received signal power

In order to use only links with acceptable quality, the EOLSR routing protocol uses two thresholds:

- a high threshold, $HighThreshold$, to accept a new link if received power $\geq HighThreshold$,
- a low threshold, $LowThreshold$, to reject a previously accepted link if received power $<$ $LowThreshold$.

The MAC layer provides to the network layer not only the received message but also the power with which the corresponding signal has been received. Hence, the EOLSR routing protocol is able to select links with good quality.

In [106], measures obtained from a MANET made of 18 routers running OLSR protocol on top of an IEEE 802.11b network are reported. These measures show that:

- the power thresholds are useful to detect mobility: the link is considered broken as soon as the received signal falls under $LowThreshold$, instead of waiting 3 times the $Hello\_Interval$ after the receipt of the last $Hello$. With the usual value of 2 seconds for the $Hello\_Interval$, this property enables a benefit of 6 seconds in OLSR reactivity.
- OLSR detects topology changes and updates routes accordingly, selecting a next hop received with an adequate received signal power.
- the variation of received power with mobility, even on some small time scale (like 10 seconds), is noticeable compared to the power variation in a static network: for instance, when computed on time intervals of 10 seconds, it can reach 6.8, whereas it is less than 3 in a static network.

Hence, the knowledge of the received signal power improves on the one hand EOLSR reactivity to topology changes by an early detection of broken links and on the other hand routes stability by the choice of good quality links.

6.3.2.2 Knowledge of the heard nodes and neighbor nodes

Without cross layering, the knowledge maintained by EOLSR results from the periodic exchange of $Hello$ and TC messages. The choice of the $Hello$ and TC periods is a tradeoff between overhead and reactivity. Any information provided by the MAC layer will be useful to increase EOLSR reactivity.

If the MAC layer of any node A receives a message from node B with a power $\geq HighThreshold$, A hears B. In the current $Hello$ period, A will send a $Hello$ message advertising that it hears B. If B receives this $Hello$ message with a power $\geq HighThreshold$, B will send a $Hello$ message in its current $Hello$ period, telling that A and B are neighbors. Hence, the exchange of two $Hello$ message is sufficient for A and B to detect that they are neighbors. In classical OLSR, three $Hello$ exchanges are needed.

We can also notice that the MAC layer is the first layer to detect the success of the attachment procedure. The attachment means that the two involved nodes have a symmetric link and hence are 1-hop neighbors. Hence these two nodes can communicate directly, this information is inserted in the routing table.

Similarly, the MAC layer is the first one to detect that a point-to-point communication between two nodes is no longer possible: no acknowledgement is received after the maximum number of retransmissions. These two nodes are now unable to communicate directly.

6.3.3 Routing and energy management cross layering

The energy management is not a layer in protocol stack. However, it can be integrated in the MAC layer or in the management device object.

The knowledge of node residual energy is needed by EOLSR to avoid nodes with low residual energy. Moreover, an energy efficient routing protocol, like EOLSR, will also select routes minimizing the energy consumed by an end-to-end transmission. These routes are built with EMPRs, Energy efficient MPRs. The selection of EMPRs avoids as much as possible nodes with low residual energy as well as nodes neighbors of nodes with low residual energy. The EMPR selection enables a graceful node energy consumption in the network.

In the next section, we show how cross layering can improve the performance of SERENA.

6.4 Cross layering with SERENA

In SERENA, a node determines when it must be awake and when it can turn off its radio. This is kept completely transparent to the routing layer, insofar as a node is awake each time a message is sent to it. However, some knowledge from the application layer and the MAC layer can improve the performance of SERENA.

6.4.1 SERENA and application layer cross layering

The knowledge provided by the application layer can help SERENA to determine when the node can sleep both from the communication and application points of view.
For these reasons, we have designed two modes of SERENA:

- generic mode described in Chapter 4: it can be applied to any type of application, where the communications are not known in advance or are very various (e.g. random source and random destination). It permits to increase the network lifetime while improving the bandwidth available to the application with the spatial reuse.

- specific mode described in Chapter 5. This solution targets data gathering applications. The benefits brought by this solution are:

  - an improvement in the network lifetime and in the bandwidth reuse (like SERENA in generic mode)
  - the delay. In fact, it assures that, in a single cycle, sensors data are collected from all sensors nodes and reported to the sink ensuring a time consistent view of these data.

In the following, we present the benefits brought by using a cross layering between SERENA and the MAC layer.

6.4.2 SERENA and MAC layer cross layering

SERENA assigns a color to each network node. The MAC layer assigns one time slot per color (it can be more than one). During a time slot, all the nodes having the corresponding color can transmit whereas their one-hop must remain awake to receive these messages. The other nodes can sleep. In order to make that possible, SERENA must notify the MAC the color of the considered node as well as the colors of its one-hop neighbors. To correctly dimension the frame, the MAC layer must know the maximum number of colors used in the network. Nodes must then behave according to the activity scheduling computed by the MAC according to the colors given by SERENA.

In addition, as we have shown in Chapter 4, If color conflicts occur, the MAC layer is the first layer able to detect this conflict. Hence, it can inform SERENA of conflicting nodes which will then correct this conflict by assigning a new color to one conflicting nodes.

6.5 Synthesis

Cross layering is a powerful approach that allows to conciliate a generic design approach and protocol optimization targeting both the desired application and the operational environment. We have first shown how to optimize the EOLSR routing protocol by:

- reducing the induced overhead by using information provided by the application. EOLSR can adapt its timers to the application phases, and maintain only routes to strategic nodes determined by the application.

- increasing EOLSR reactivity by using information provided by the MAC layer: (e.g. received signal power, knowledge of heard nodes).

- improving energy efficiency by (i) avoiding nodes with low residual energy, (ii) selecting routes minimizing the consumed energy.

Then, we have presented the different cooperations between SERENA and on the one hand the application layer and on the other hand the MAC layer. They allow:

- SERENA to adapt to different types of applications and their requirements. It can be applied in generic mode to any application. It will improve the network lifetime and the bandwidth by means of the spatial reuse. SERENA can be also adapted to specific requirements of some applications like data gathering. In such a case, it will be used in specific mode. It will decrease the delay of data collection or dissemination in the network.

- SERENA to solve color conflicts that can occur after execution of the coloring algorithm due to the creation of new links resulting from node mobility, late node arrival or topology changes. To achieve that, a detection of these conflicts must be done in MAC layer which informs SERENA. SERENA will then known all the information needed to solve conflicts.

We will now present the OCARI project for which we have designed and specified an energy efficient network layer including EOLSR and SERENA. We will also describe the cross layering between these two modules and the MAC and application layers.

6.6 Application to the OCARI project

6.6.1 Project description

OCARI[111], (Optimization of Communication for Ad hoc Reliable Industrial networks), is a wireless communication technology targeting applications in harsh environments such as power plants and warships. It supports mesh topology and provides a deterministic MAC access for time-constrained communications as well as energy efficient communications for an increased network lifetime. Based on the PHY layer of IEEE 802.15.4, it supports the ZigBee APS and APL primitives and profiles, ensuring compatibility with existing applications.

The OCARI project, funded by the French National Research Agency and industrial partners. It started at the end of 2006 and gathers partners from industries (EDF/project leader, DCNS and Telit) as well as university labs and research institutes (LIMOS, LATTIS, LRI and INRIA). EDF and DCNS provide requirements and use cases in power industry and warship applications. Telit, a high tech company in wireless communication will industrialize the prototype. LIMOS and

LATTIS university laboratories develop and implement OCARI medium access methods. LRI in charge of the energy model to compute the residual energy in nodes. INRIA works on optimized energy consumption in the network.

6.6.1.1 Goals of OCARI

OCARI focuses on developing a complementary industrial specification. It aims at responding to the following requirements which are particularly important in power generation industry and in warship construction and maintenance:

- Support of deterministic MAC layer for time-constrained communication,
- Support of optimized energy consumption routing strategy in order to maximize the network lifetime,
- Support of human walking speed mobility for some particular network nodes, (e.g. sinks),
- Support of IEC61804/EDDL and HART application layer [110].

The development of OCARI targets the following industrial applications:

- Real time centralized supervision of personal dose in nuclear power plants,
- Condition Based Maintenance of mechanical and electrical components in power plants as well as in warships,
- Environmental monitoring in and around power plants,
- Structure monitoring of hydroelectric dams. We now present the OCARI network.

6.6.2 The OCARI network and its architecture

6.6.2.1 Topology of an OCARI network [99]

The topology of an OCARI network can be modeled as depicted in Figure 6.3. An OCARI End Device is a "radio fixed" network node (i.e. its position varies very little comparing to its initial location so that its radio link is always managed by the same cell coordinator). It is a Reduced Function Device as defined in IEEE 802.15.4 specification. OCARI Cell and Workshop Coordinators are Full Function Devices as specified in IEEE 802.15.4. They are fixed devices in the infrastructure and are equivalent to ZigBee Coordinators. The functions of Cell Coordinator consist of:

- Coordinating the intra-cell network nodes using a star topology, and
- Routing data packets in push mode from end device network nodes (i.e. sensors in the industrial applications) to the Workshop Coordinator per Workshop domain. Tree routing is used in time-constrained period and EOLSR is used otherwise between Cell Coordinators.

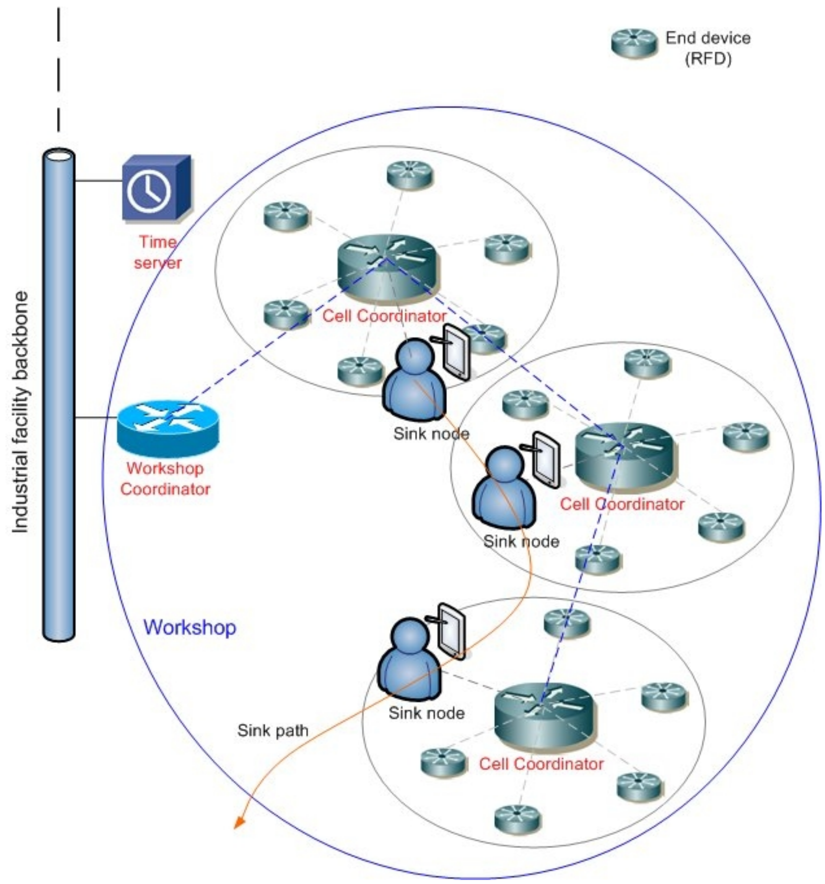

Figure 6.3: Topology of an OCARI network.

Workshop domain is a permissive volume (delimited by a threshold of the BER) to electromagnetic wave that is covered by a unique Workshop Coordinator network node. Workshop Coordinator is a gateway between the wireless sensors network resided in a Workshop domain and the industrial facility backbone. As depicted in Figure 6.3, Sink node is a mobile network node which usually represents a patrolman/maintenance operator, equipped with PDA, collecting data from sensors inside a cell. Sink leaves the cell when it finishes to acquire data (in polling mode). Time server is a particular network node which is used for clocks synchronization in the whole Workshop domain.

6.6.2.2 OCARI architecture [99]

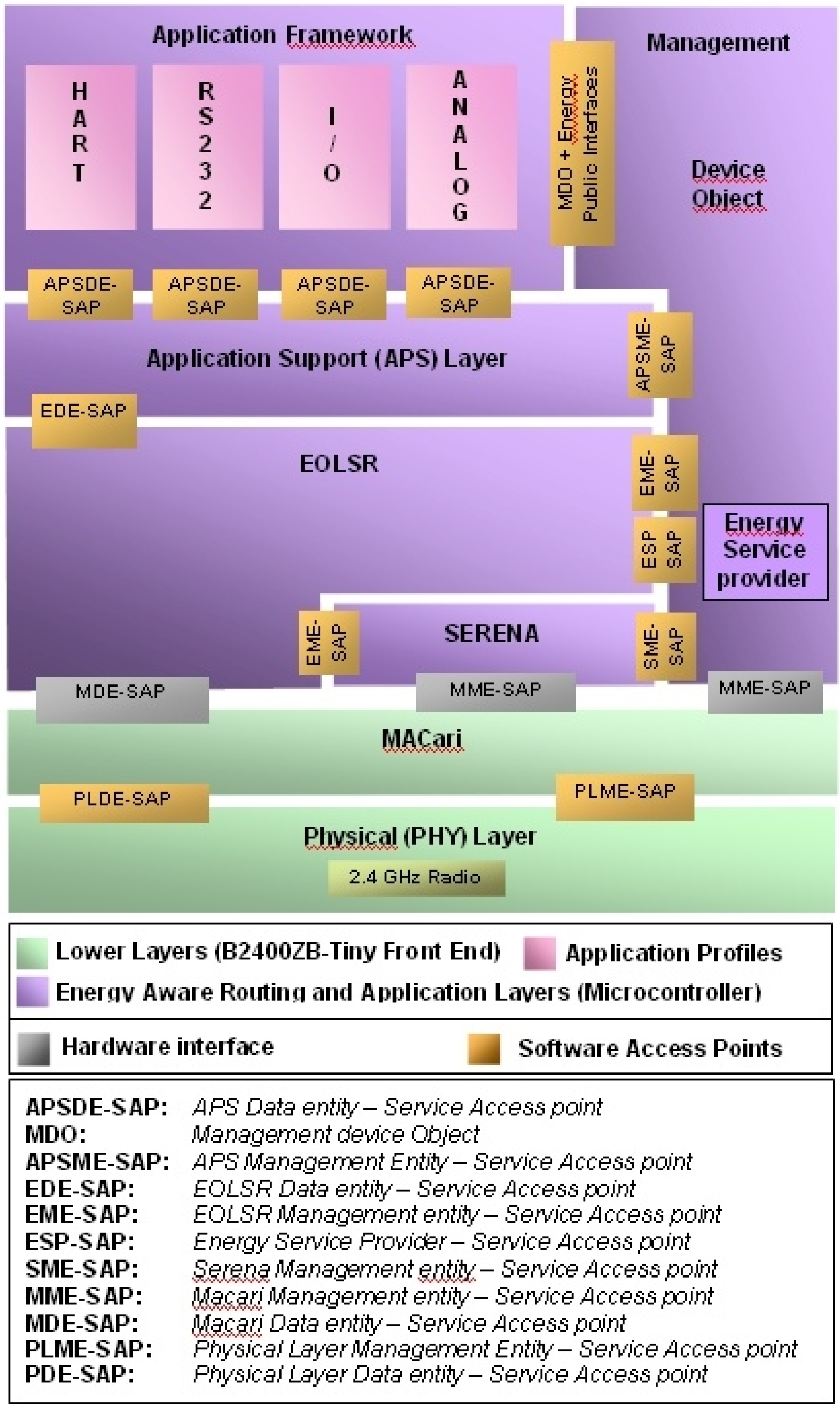

Figure 6.4: OCARI architecture.

Figure 6.4 depicts the OCARI architecture of a network node with the interfaces between the different layers. Notice the Energy Service provider module managing the residual energy. We now detail these layers.

6.6.2.3 MACARI: the MAC layer [99]

MaCARI [107, 108] is the MAC protocol used in OCARI and designed by the LIMOS and the LATIIS teams. It is based on a tree topology rooted at the Workshop coordinator and has several characteristics: flexibility, determinism and low energy consumption. Its flexibility is obtained through the means of three repetitive time periods constituting a global cycle that can be redefined at the start of every cycle (see Figure 6.5).

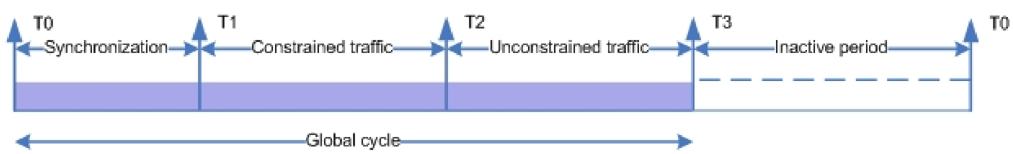

Figure 6.5: Global cycle.

The first period ($[T_0, T_1]$) corresponds to a synchronization broadcasted hop by hop from the Workshop coordinator to all the entities in the network. The second period ($[T_1, T_2]$) corresponds to the collection of data from the sensors, each cell being activated sequentially. During this period, preallocated time slots are reserved for inter-coordinator communications that follow the tree topology, that is why we refer to such communications as deterministic. In other words, each coordinator has a dedicated period of time to exchange with its parent and with its child. The third period ($[T_2, T_3]$) corresponds to unconstrained communications between coordinators as long as they are in range. During this period, packets are routed according EOLSR routing protocol. During these three periods, MaCARI makes network entities sleep as often as possible, saving energy.

The activity of each cell is managed by slotted CSMA/CA. To provide determinism, a mechanism of guaranteed time-slots (GTS) can be used, as MaCARI guarantees that neighboring cells do not interfere with the activated cell. In comparison with the original IEEE 802.15.4 stack, MaCARI also integrates different types of GTSs: GTS(n) with multiple reservation levels (one GTS is allocated every n superframes, enabling different levels of QoS), GTS can be preallocated, or GTSs can even be simultaneous (under restricting conditions).

6.6.2.4 Energy service provider: Residual energy computation [99]

Before describing these strategies used in OCARI, we now present the energy model used to compute the residual energy of a node. This model is developed by the LRI team. This model estimates the residual energy of the node. This residual energy is used by EOLSR to select the

EMPR (Energy efficient MPRs).

The goal in OCARI is to integrate the different functions done by a node in order to estimate its energy level. The energy level will serve as a metric for the routing protocol to select its EMPRs, intermediates nodes in computed routes. The model is described as follows: each node is affected an initial value of energy and an amount of energy is subtracted according to its current operation (transmission, reception, wake up ...) and also to the date of this operation. The date takes into account how many times the battery was used. The more the battery is used, the less is its energy. An exponential function where the slope corresponds to the date and the amplitude to the energy consumed by the specific operation is defined: $R_r(0) = Max$ and $R_r(i) = R_r(i-1) - c(i) \times e^{-const \times \frac{\Delta}{i}}$, with:

- $R_r(i), i \in N$ is the remaining energy in a node at date i
- Max is the maximum energy for a node
- $c(i)$ is the consumed energy during step i
- Δ is the duration of step i
- $const$ is a constant

The value of $c(i)$ is calculated according to the ZigBee standard, where:
• $E_{tx} = 30.2 \times 10^{-6} mAH + 8.0556 \times 10^{-6} \times t_1$ for packet transmission,
• $E_{rx} = 40.986 \times 10^{-6} mAH + 8.0556 \times 10^{-6} \times t_1$ for packet reception.

6.6.2.5 Application layer [99]

To achieve a seamless integration of wireless sensors network (WSN) into real world applications in industrial information systems, we need to develop and provide an application architecture that can interoperate with existing industrial standards. The architecture depicted in Figure 6.6 aims at responding to such requirement while supporting the state of the art in industrial information technology.

The role of a WSN oriented middleware is to provide standard and homogenous services to user applications. It also contributes to the energy saving of the WSN by implementing a centralized and optimized management of network resources. Moreover, it serves as a gateway between different user applications and different Workshop Coordinators, which orchestrate the attached cells. The choice of a software bus standard such as OPC-DA and the upcoming version OPC-UA allows us to provide the compatibility with EDDL for asset management and for SCADA application.

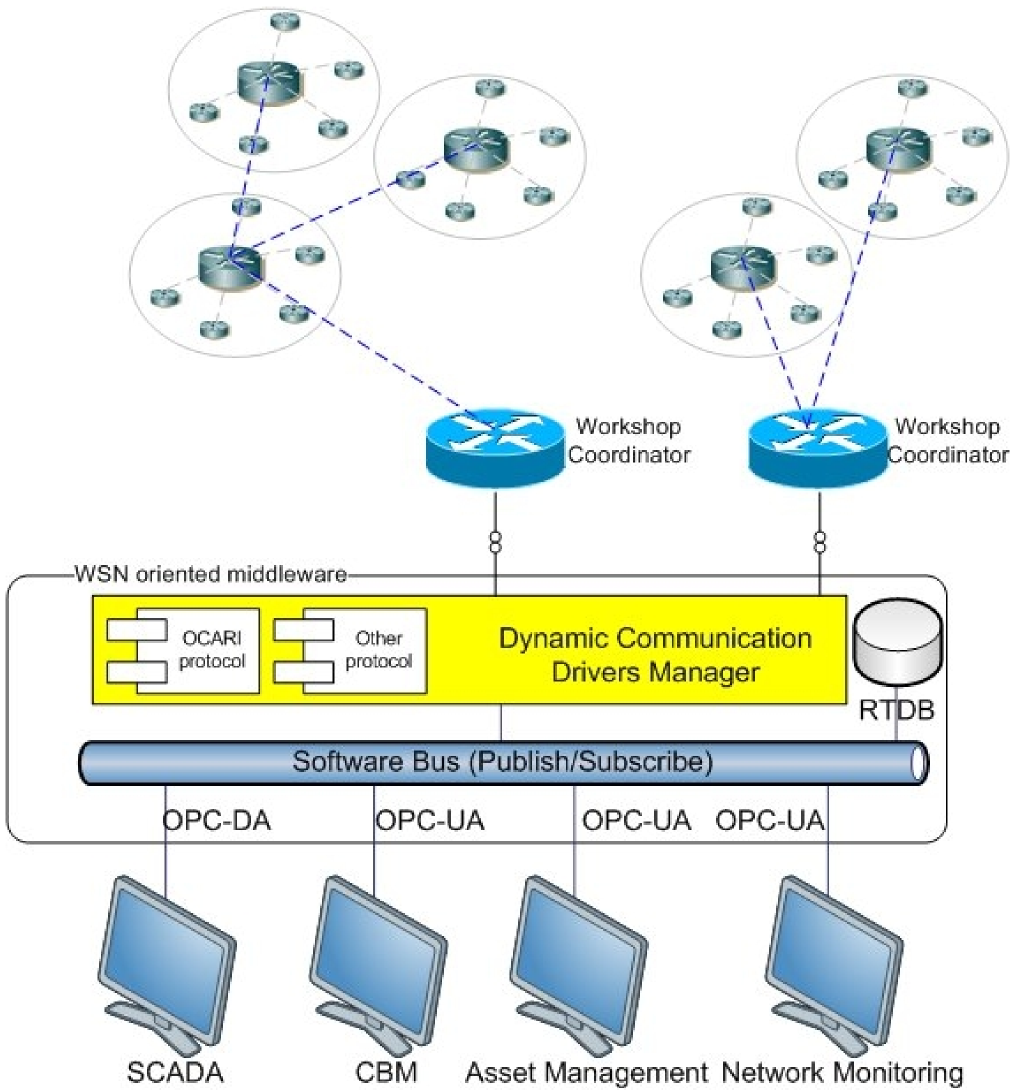

Figure 6.6: OCARI application architecture.

6.6.3 NwCARI: Energy efficient network layer

NwCARI is the network layer of OCARI. We were in charge of this layer, and particularly of the specification of the energy efficient network layer of OCARI.

6.6.3.1 Services offered

This network layer allows:

- an end-to-end communication while maximizing the lifetime of the network by:
  - adaptive and energy aware routing: EOLSR.
  - Node activity scheduling: SERENA.

- Supports two types of traffic:
  - constraint traffic. This traffic has a delay constraint. It is relayed by the MaCARI layer in the tree. This traffic is relayed in $[T_1, T_2]$ interval.
  - unconstrained traffic. It is routed by the routing layer using routes built by EOLSR. This traffic has no delay constraint. This traffic is routed in $[T_2, T_3]$ interval. Fragmentation is allowed for unicast unconstrained traffic.
- Supports node mobility. This mobility is limited to some nodes in the network.
- Broadcast optimization. The broadcast optimization is offered by using the MPRs selected by EOLSR for sending broadcast messages. These messages are messages generated by EOLSR (i.e. TCs messages) or user broadcast messages.

Our target is to design the network layer to support all these services. To achieve that, we have specified:

- the network messages format: broadcast and unicast user packet and control packet of SERENA and EOLSR. The size of a frame in OCARI is limited to 128 bytes. This small size adds constraints when designing messages format. To solve this problem, we have specified a message fragmentation and reassembling in the network layer. For more details,the reader can refer to [100].
- the primitives needed in network layer to communicate with the application layer and the MAC layer. For more details,the reader can refer to [101].
- the energy efficient routing protocol: EOLSR.
- the node activity scheduling SERENA: which allows nodes to sleep to spare energy while assuring the end-to-end communication.

In the following, we present the design choices taken in OCARI to reach our target.

6.6.3.2 Design choices

The energy constrained nature of OCARI nodes requires the use of energy efficient strategies to maximize network lifetime. To achieve that we have designed EOLSR and SERENA. These two protocols are used in the third period defined by MaCARI $[T_2, T_3]$, which corresponds to unconstrained communications between coordinators.

EOLSR: energy efficient routing protocol.
We have presented, in Chapter 3, EOLSR the energy aware routing protocol based on OLSR. This protocol answers two OCARI objectives. The first one is the energy efficiency. In fact, EOLSR builds routes that minimize the end-to-end energy consumption using nodes with high residual energy as

intermediate nodes. The second objective is supporting human walking speed mobility for some particular network nodes.
Moreover, EOLSR operates in two modes depending on the application requirements:

- A generic mode where each node maintains a route to any other node in the network.
- A strategic mode where any non-strategic node maintains only a route to each strategic node (sink node). It is made possible by cross-layering between the application and the routing layers.

These two modes can be used in OCARI with regards to the application requirements.
Finally, a cross layering between EOLSR and the energy service provider entity which computes the residual energy of nodes, makes possible the computation of EPMRs.
The EOLSR protocol is under implementation by the industrial partner Telit on ZigBee platform.

SERENA: node activity scheduling.
Concerning SERENA, presented in Chapter 4 for generic applications and in Chapter 5 for data gathering applications, it is used to improves the energy efficiency in OCARI. SERENA works conjunctively with MaCARI:

- SERENA is in charge of assigning colors to nodes. At the end of the coloring phase, a MaxColor message is sent by SERENA to inform the MaCARI of workshop coordinator about the max value color used in the network. This color is used to fix the number of slots in the interval $[T_2, T_3]$; each slot corresponds to a color. Moreover, SERENA sends the color of the node and the color of the one-hop neighbors to the local MaCARI. Consequently, knowing the number of colors used in the network and the color of the node and its one-hop neighbors, MaCARI can schedule node activity. Hence, it must be awake in its slot (corresponding to the color of the node) to send messages and in the slots of its one-hop neighbors (corresponding to the colors of its one-hop neighbors) to receive messages destinated to itself.
- As we have shown in Section 4.5, Chapter 4, color conflict can occur. This conflict should be solved locally. To achieve that, a cross layering with MaCARI layer is introduced. In fact, the MAC layer uses a specific medium access mechanism to detect unexpected behaviors and report information concerning nodes in conflict to SERENA which solves this conflict. The method used by MaCARI to detect these conflict named TDMA/CA is presented in Section 4.5.2, of Chapter 4.

SERENA and primitives necessary to exchange information between the MAC layer are under implementation.

6.6.3.3 Benefit of the cross layering combining energy efficient routing and node activity scheduling

We now quantify the benefits brought by EOLSR and SERENA used separately or in combination by evaluating the network lifetime. As the MaCARI layer is under implementation, the results shown in Figure 6.7 have been obtained with an IEEE 802.11 network at 2Mbps, a network density of 10, an initial node energy of 100 Joules. User traffic consists of 30 CBR flows, with randomly chosen sources and destinations, a throughput of 16 Kbps and a message size of 512 bytes.

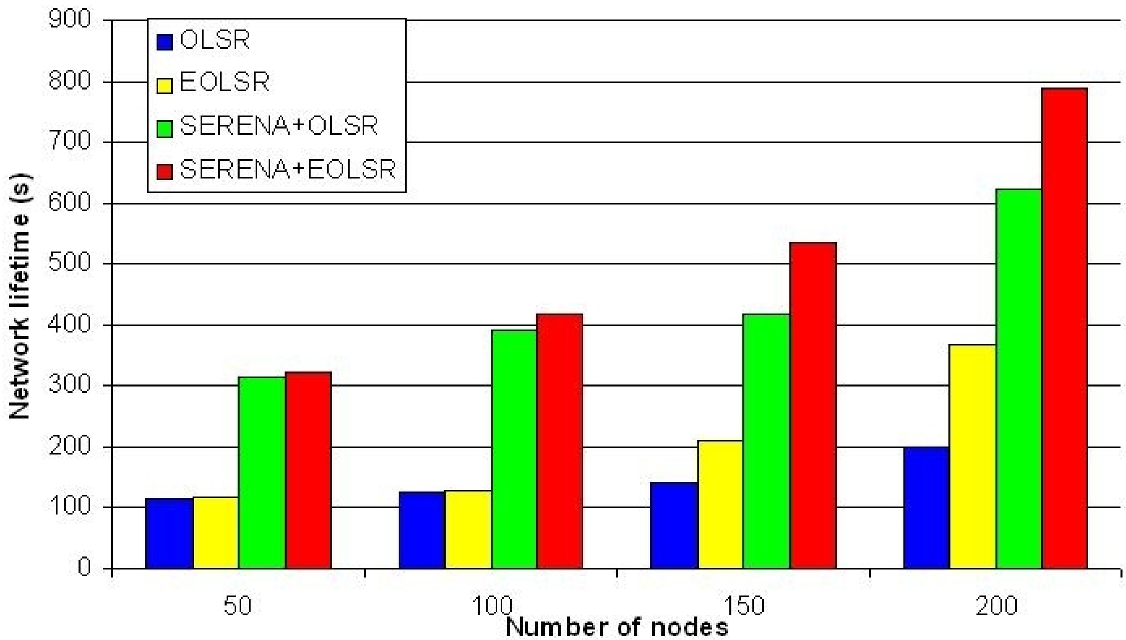

Figure 6.7: Network lifetime.

Figure 6.7 shows that EOLSR increases the network lifetime of about 40% with regard to OLSR in a network of 150 nodes. This is due to the choice of energy efficient routes. However, the major improvement is brought by SERENA that ensures a better use of the bandwidth, limited in wireless sensor networks. All routers colored with the same color can transmit simultaneously without collision and without interference, leading to a good spatial reuse. EOLSR and SERENA, used separately or in combination, contribute to a more efficient use of node energy and hence a higher network lifetime.

We are working now on the implementation and the development of different components of our specification. The integration of the different OCARI layer will be done. To fully assess the OCARI technology, extensive tests will be done in industrial facility at EDF R&D as well as at DCNS.

6.7 Conclusion

In this chapter, we have shown how the cross layering can improve the network performances. We have first shown how to optimize the EOLSR routing protocol by a cooperation with the application layer (to reduce the overhead) and with the MAC layer (to increase the reactivity of the routing protocol). Then, we moved on to the cross layering between SERENA and the application layer to meet to its requirements (minimize the energy consumption, improve the bandwidth and decrease the delay for data gathering applications) and with the MAC layer to allow SERENA to solve color conflicts for example. In a second step, we have presented the OCARI project in which our work has been integrated. Our contribution consisted in designing the energy efficient network layer of OCARI (NwCARI). This means to define the different services offered by this layer and so specify: (1) messages formats, (2) different primitives, (3) how this layer can be energy efficient: EOLSR and SERENA are integrated in this layer to improve the energy efficiency of OCARI. The NwCARI is under implementation.

Chapter 7

Conclusion and perspectives

7.1 Conclusion

The diversity of the applications supported by wireless ad hoc and sensor networks explains the success of this type of network. These applications include both military as well as civilian applications. However, one of the most critical design issues of wireless ad hoc and sensor networks is the limited amount of nodes' energy. As energy may be difficult to renew in such networks, protocols running on these networks must be energy efficient.
In our doctoral work, we have addressed this problem, when our first target is to prolong the network lifetime. To achieve that, we have first classified these researches in four classes: 1) energy efficient routing, 2) node activity scheduling, 3) reducing the amount of information transferred, and 4) topology control. Then, we have highlighted the sources of wasted energy in a wireless node.

Lesson 1: To achieve the best energy efficiency, we need to combine several energy efficient strategies.

In this dissertation, we focused on the first three classes of energy efficient strategies to maximize the network lifetime. First, we have focused on the energy efficient routing. We have extended the proactive routing protocol OLSR to take into account the energy. EOLSR, Energy efficient OLSR, is proposed. It consists of four modules: (1) energy consumption model, (2) energy efficient selection of multipoint relay named EMPR, (3)energy efficient routing algorithm and (4) optimization broadcasts. The originality of this protocol resides in the computation of routes that minimize the energy consumption from nodes with sufficient residual energy (nodes selected as EMPR). Moreover, EOLSR aims at balancing load charge in the network by changing the EMPR selection. We have implemented EOLSR on the network simulator ns2. Comparative performance evaluation results show that EOLSR outperforms, on the one hand, the $MinEnergy$ and $MaxPacket$ that respectively minimizes the end-to-end energy consumption and maximizes the number of packets sent. On the other hand, EOLSR outperforms the two variants of multipath routing (with different links and with different nodes). A cross layering between EOLSR and both the MAC layer and the application layer is used to improve its performances. A cooperation between EOLSR and the application layer allows us to define two modes of EOLSR:

- generic mode where each node maintains a route to any other node in the network. This mode is applied to applications where the communications are not known in advance.
- strategic mode where any non-strategic node maintains only a route to each strategic node. it is applied to applications where the potential destinations exist in a limited number and are known in advance. This is the case for instance in data gathering applications. This mode considerably reduces the overhead of EOLSR.

In conclusion, we point out that the EMPR selection contributes to energy balancing between nodes. The route computation based on minimum energy consumption does not tend to select long routes. Indeed, at each hop, we add the energy consumed in transmission, reception and overhearing. However, the main source of optimization(i.e. high reduction of the overhead induced in bandwidth and storage) resides in the cross layering with the application where the strategic mode can used. This optimization is very useful in wireless networks with resources of limited capacity.

Second, we have been interested in node activity scheduling since the energy consumed in the sleep state is lesser than the one consumed in the transmit, receive and idle states. SERENA, SchEduling RoutEr Node Activity, is proposed to maximize the network lifetime in wireless ad hoc and sensor networks. This solution allows router nodes to sleep, while ensuring the network functionality and the end-to-end communication. SERENA consists in two distributed and localized algorithms: a coloring algorithm and a slot assignment algorithm based on colors. To design SERENA, we have first classified the different application requirements and network environment constraints. Then, we have shown how we can extend SERENA to take into account these conditions. We have presented how SERENA deals with immediate acknowledgement of unicast transmissions. Then, we have pointed out that the existence of unidirectional links, node mobility and the late node arrivals can lead to color conflicts. We have described the different types of conflicts and have evaluated their frequency. To solve these conflicts, we have proposed a cross layering between SERENA and the MAC layer which is in charge of detecting the conflict, extracting the addresses of the conflicting nodes and sending this information to SERENA which corrects this conflict.

Lesson 2: Graph theory algorithms assuming ideal network conditions cannot be applied in real wireless ad hoc and sensor networks.

We have then improved the matching of SERENA with application requirements. In fact, improving bandwidth and end-to-end delay guarantees in a wireless sensor network for data gathering applications while maximizing the network lifetime is a crucial problem. We have proposed an extension of SERENA, ensuring that data gathering or data dissemination takes only a single cycle. This solution also ensures time consistency of collected data. Hence, a cross layering with the application layer permits us to select the most appropriate mode of SERENA:

- generic mode is used where the is no specific knowledge on the application and topology.
- specific mode is used in application organized in a tree like data gathering applications

In conclusion on SERENA, we have proposed a generic approach that can be adapted to various application constraints. Here also, we have shown the high benefits brought by cross layering with the application. The end-to-end delay of data gathering can be reduced to single polling cycle.

Third, we have shown how to optimize broadcasts by means of multipoint relays in a wireless sensor network in order to prolong network lifetime. This optimization belongs to the third energy efficient strategy, namely reduction of the amount of data transferred.

Lesson 3: The best improvement of energy efficiency is obtained by cross layering.

This doctoral work has been applied in OCARI project, to design an industrial wireless sensor networks. In this project, we were in charge of designing the energy efficient network layer. We have specified network messages format, network primitives and processing done by the network layer. We have integrated in the OCARI network layer the three modules we have designed namely: EOLSR, SERENA, and optimized broadcasts. we have taken advantages of cross layering with the MAC and application layers. This layer is under implementation. The participation in this project was a good experience and allowed to us to take into account real industrial environment constraints in our work.

Lesson 4: Real industrial environment constraints are a powerful incentive to design energy efficient protocols.

7.2 Perspectives

Energy efficiency in wireless ad hoc and sensor networks is an active research area. In the following we show how our work can be extended and expanded in various research directions.

- The first idea as an extension to our work is to exploit the cross layering more strongly to improve the performances of our algorithms. We can capture more information from the application layer to optimize the communication protocols and node activity scheduling to meet the application requirements. We will study how to tune the EOLSR routing parameters in order to meet the application requirements while minimizing the induced overhead.
- In Chapter 3, we have shown that color conflicts are unavoidable. Hence, designing a color conflict detection and resolution mechanism is necessary to the good performance of SERENA. However, we should envisage the recoloring of the network. When do we decide to recolor all the network instead of detect and solve color conflicts?

- In our work, we have used simulation tools to evaluate the performance of our routing protocol and the number of colors and rounds for SERENA. In a second stage, it will be interesting to design a mathematical model for our protocol to evaluate theoretical bounds on the number of colors and rounds.

- Our work can help the designer to build its wireless sensor network and to dimension it. We have shown how to compute the end-to-end delay in a data gathering application as well as the buffer size at each node. It would be interesting to prolong this work by designing a dimensioning tool for wireless sensor networks.

- We will extend SERENA to be applied for other targets. For instance, we can study the possible use of SERENA coloring algorithm to assign frequencies to overlapping set of nodes. For example, if several PANs (Personal Area Networks) coexist, we can use SERENA to assign a color per PAN. This color will then be mapped in a frequency.

- Another perspective is the test of the deployment of our solution in industrial wireless sensor networks. The OCARI project was an excellent opportunity for us to do that. We now wish to experiment our solutions in larger scale. Tools to assist in network deployment would be very useful.

- With regard to standardization it is needed to position our work concerning EOLSR with regard to RPL (Routing Protocol for Low Power and Lossy Networks draft-ietf-roll-rpl-03), a draft presented at IETF. It seems that RPL is a framework that allows designers to develop solutions taking into account various criteria including energy. Whereas EOLSR is centered on energy efficiency.

Publication

- **International Journals**

1. S. Mahfoudh, P. Minet, *EOLSR: an energy efficient routing protocol in wireless ad hoc and sensor networks*, Journal of Interconnection Networks, JOIN, Special Issue in Routing, Scheduling & Load balancing in Networking Systems, December 2008.
2. S. Mahfoudh, P. Minet, *Maximization of energy efficiency in wireless ad hoc and sensor networks with SERENA*, Journal on Mobile Information System (MIS) 2009.
3. K. Agha, G. Chalhoub, A. Guitton, E. Livolant, S. Mahfoudh, P. Minet, M. Misson, J. Rahmé, T. Val, A. Van Den Bossche, *Cross-layering in an industrial wireless sensor network: case study of OCARI*, Journal of Wireless Networks 2009.

- **International Conferences**

4. S. Mahfoudh, P. Minet, *Performance Evaluation of the SERENA Algorithm to SchEdule RoutEr Nodes Activity in Wireless Ad Hoc and Sensor Networks*, AINA 2008, Ginowan, Japan, March 2008.
5. S. Mahfoudh, P. Minet, *An energy efficient routing based on OLSR in wireless ad hoc and sensor networks*, PAEWN 2008, Ginowan, Japan, March 2008.
6. S. Mahfoudh, P. Minet, *Survey of energy efficient strategies in wireless ad hoc and sensor networks*, ICN 2008, IEEE International Conference on Networking, Cancun, Mexico, April 2008.
7. P. Minet, S. Mahfoudh, *SERENA: SchEduling RoutEr Nodes Activity in wireless ad hoc and sensor networks*, IWCMC 2008, Crete Island, Greece, August 2008.
8. M. Bertin, A. Bossche, G. Chalhoub, T. Dang, S. Mahfoudh, J. Rahmé, J. Viollet *OCARI for industrial wireless sensor networks*, IFIP Wireless Days 2008, Dubai, United Arab Emirates, November 2008.
9. P. Minet, S. Mahfoudh, *Cross layering in wireless sensor networks*, IWCMC 2009, Leipzig, Germany, June 2009.
10. S. Mahfoudh, P. Minet *Node activity scheduling in wireless sensor networks*, RTNS 2009, Paris, France, October 2009.
11. P. Minet, S. Mahfoudh, *Energy, bandwidth and time efficiency in data gathering applications*, IFIP Wireless Days 2009, Paris, France, December 2009.

12. P. Minet, S. Mahfoudh, G. Chalhoub, A. Guitton, *Node coloring in a wireless sensor network with unidirectional links and topology changes*, To appear in WCNC 2010, Sydney, Australia, April 2010.

Bibliography

[1] R. C. Shah, *Distributed algorithms to maximize the lifetime of wireless sensor networks*, PhD Thesis, University of California, Berkeley, June 2005.

[2] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, *Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks*, IEEE Transactions on Parallel and Distributed Systems, June 2006.

[3] M. Cardei, J. Wu, S. Yang, *Topology Control in Ad hoc Wireless Networks with Hitch-hiking*, First IEEE SECON04, October 2004.

[4] K. Kalpakis, K. Dasgupta, P. Namjoshi, *Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks*, IEEE Networks'02, Munich, Germany, August 2002.

[5] X. Tang, J. Xu, *Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks*, IEEE INFOCOM 2006, Barcelona, Spain, April 2006.

[6] J. Kulik, W. Rabiner, H. Balakrishnan, *Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*, 5th ACM/IEEE Mobicom Conference, Seattle, WA, August 1999.

[7] M. Cardei, M. Thai, Y. Li, W. Wu, *Energy-efficient target coverage in wireless sensor networks*, IEEE INFOCOM 2005, Miami, Florida, March 2005.

[8] J. Carle, D. Simplot-Ryl, *Energy-Efficient Area Monitoring for Sensor Networks*, IEEE Computer Magazine, vol. 37, no. 2, pp. 40-46, Febuary, 2004.

[9] M. Cardei, D. Du, *Improving wireless sensor network lifetime through power aware organization*, ACM Journal of Wireless Networks, May 2005.

[10] W. Ye, J. Heidmann, D. Estrin, *An Energy-Efficient MAC Protocol for Wireless Sensor Networks*, IEEE INFOCOM, New York, USA, June 2002.

[11] T. V. Dam, K. Langendoen, *An adaptive energy-efficient MAC protocol for wireless sensor networks*, ACM SenSys'03, November 2003.

[12] G. Lu, B. Krishnamachari, C. Raghavendra, *An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks*, Parallel and Distributed Processing Symposium, April 2004.

[13] G. Lu, B. Krishnamachari, C. Raghavendra, *O-MAC: An Organized Energy-Aware MAC Protocol for Wireless Sensor Networks*, IEEE ICC, Glasgow, UK, June 2007.

[14] The Institute of Electrical and Electronics Engineers, Inc., *IEEE Std 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1999 edition.

[15] P. Minet, *Energy efficient routing*, Book chapter, Ad Hoc and Sensor Wireless Networks: Architectures, Algorithms and Protocols published by Bentham Science, 2009.

[16] C. Intanagonwiwat, R. Govindan, D. Estrin, *Directed Diffusion: a scalable and robust communication paradigm for sensor networks*, MobiCom 2000, Boston, MA, August 2000.

[17] A. Manjeshwar, D.P. Agrawal, *TEEN: a protocol for enhanced efficiency in wireless sensor networks*, Workshop on Parallel and distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[18] A. Manjeshwar, D.P. Agrawal, *APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, Workshop on Parallel and distributed Computing Issues in Wireless Networks and Mobile Computing, Fort Lauderdale, FL, April 2002.

[19] Y. Yu, R. Govindan, D. Estrin, *Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks*, UCLA Computer Science Department UCLA-CSD TR-01-0023, May 2001.

[20] Y. Xu, J. Heidemann, D. Estrin, *Geography-informed energy conservation for ad hoc routing*, MobiCom 2001, Rome, Italy, July 2001.

[21] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, *SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks*, MobiCom 2001, Rome, Italy, July 2001.

[22] G. Girling, J. Li, K. Wa, P. Osborn, *The design and implementation of a low power ad hoc protocol stack*, IEEE Personal Communications, 2000.

[23] K. Akkaya, M. Younis, *A survey on routing protocols for wireless sensor networks*, Ad Hoc Networks Journal, Vol3, 2005.

[24] M. Liu, J. Cao, G. Chen, X. Wang, *An energy-aware routing protocol in wireless sensor networks*, Sensors Vol.9, 2009.

[25] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, *Energy-efficient communication protocol for wireless microsensor networks*, HICSS 2000, Maui, Hawaii, USA, vol. 2, January 2000.

[26] D. Xia, N. Vlajic, *Near-optimal node clustering in wireless sensor networks for environment monitoring*, AINA 2007, Niagara Falls, Ontario, Canada, May 2007.

[27] R.C. Shah, J.M. Rabaey, *Energy Aware Routing for Low Energy Ad Hoc Sensor Networks*, IEEE WCNC 2002, Volume 1, pp. 17-21, March 2002.

[28] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, *Highly-resilient,energy-efficient multipath routing in wireless sensor networks*, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), volume 1, no 2, 2001.

[29] A. Srinivas, E. Modiano, *Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks*, MOBICOM'2003, September 2003.

[30] S.-M. Senouci, G. Pujolle, *Energy efficient routing in wireless ad hoc networks*, ICC 2004, volume 7, pp. 4057-4061, June 2004.

[31] S. Kwon, Ness B. Shroff, *Energy-Efficient Interference-Based Routing for Multi-hop Wireless Networks*, IEEE INFOCOM'2006, Barcelona, Spain, April 2006.

[32] H. Hassanein, Jing Luo, *Reliable Energy Aware Routing In Wireless Sensor networks*, IEEE DSSNS 2006, April 2006.

[33] N.Shresta, *Reception Awarness for Energy Conservation in Ad Hoc Networks*, PhD Thesis, Macquarie University Sydney, Australia, November 2006.

[34] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, *Optimized Link State Routing Protocol*, RFC 3626, IETF, 2003.

[35] T. Plesse, C. Adjih, P. Minet, A. Laouiti, A. Plakoo, M. Badel, P. Mühlethaler, P. Jacquet, J. Lecomte, *OLSR performance measurement in a military mobile ad-hoc network*, Ad Hoc Networks Journal, Elsevier, Vol 3/5, September 2005.

[36] B. Mans, *On the Complexity of Reducing the Energy Drain in Multihop Ad Hoc Networks*, INRIA research report, RR-5152, INRIA Rocquencourt, France, March 2004.

[37] G. Allard, P. Minet, D.Q. Nguyen, N. Shresta, *Evaluation of the energy consumption in MANET*, Adhoc-Now 2006, Ottawa, Canada, August 2006.

[38] D.Q. Nguyen, P. Minet, *Analysis of MPR selection in the OLSR protocol*, PAEWN 2007, Niagara Falls, Ontario, Canada, May 2007.

[39] A. Nasipuri, S. Das, *On-demand multipath routing for mobile ad hoc networks*, Int. Conf. on Computer Communications and Networks, Boston, MA, October 1999.

[40] G. Toussaint, *The relative neighborhood graph of a finite planar set*, Pattern Recognition, Vol 12, 1980.

[41] J. Wieselthier, G. Nguyen, A. Ephremides, *On the construction of energy-efficient broadcast and multicast trees in wireless networks*, Infocom 2000, Tel Aviv, Israel, March 2000.

[42] N. Li, J. Hou, L. Sha, *design and analysis of an MST-based topology control algorithm*, Infocom 2003, San Francisco, CA, April 2003.

[43] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, *ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks*, SenSys'06, Colorado, November 2006.

[44] G. Pei, M. Gerla, T.-W. Chen, *Fisheye state routing: a routing scheme for ad hoc wireless networks*, IEEE ICC'00, New Oreleans, LA June 2000.

[45] D.Q. Nguyen, P. Minet, *Scalabilty of the OLSR protocol with the Fish Eye extension*, ICN 2007, Sainte Luce, Martinique, April 2007.

[46] P. Jacquet, A. Laouiti, P. Minet, L. Viennot, *Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models*, INRIA Research Report, RR 4260, http://www.inria.fr/rrrt/rr-4260, September 2001.

[47] J. Wu, H. Li, *On calculating connected dominating set for efficient routing in ad hoc wireless networks*, ACM DIAL M, Seattle, Washington, August 1999.

[48] F. Dai, J. Wu, *An extended localized algorithm for connected dominating set formation in ad hoc wireless networks* IEEE Trans. on Parrallel and distributed systems, 15(10), 2004.

[49] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, *Smaller Connected Dominating Sets in Ad Hoc and Sensor Networks based on Coverage by Two-Hop Neighbors*, Proc. 2nd International Conference on Communication System Software and Middleware, Bangalore, India, 2007.

[50] P. Jacquet, *Analytical results on connected dominating sets in mobile ad hoc networks* INRIA Research Report, RR 5173, http://www.inria.fr/rrrt/rr-5173, April 2004.

[51] K. Alzoubi, P.J. Wan, O. Fieder, *Distributed heuristics for connected dominating sets in wireless ad hoc networks* Jal of Communications and Networks, 4(1), March 2002.

[52] B. Han, H.H. Fu, L. Li, W. Jia, *Efficient construction of connected dominating set in wireless ad hoc networks*, IEEE MASS 2004, Fort Lauderdale, Florida, October 2004.

[53] J. Zhu ,S. Chen, B. Bensaou, K. Hung, *Tradeoff Between Lifetime and Rate Allocation in Wireless Sensor Networks: A Cross Layer Approach*, IEEE INFOCOM 2007, Alaska, USA, May 2007.

[54] H. Nama, M. Chiang, N. Mandayam, *Optimal Utility-Lifetime Trade-off in Self-regulating Wireless Sensor Networks Ũ A Distributed Approach*, CISS 2006 Princeton,NJ, USA, Mars 2006.

[55] H. Nama, M. Chiang, N. Mandayam, *Utility-Lifetime Trade-off in Self-regulating Wireless Sensor Networks Ũ A Cross-layer Design Approach*, IEEE ICC 2006, Istanbul, Turkey, May 2006.

[56] H. Kwon, T. Kim, S. Choi, B.G. Lee, F *A Cross-Layer Strategy for Energy-Efficient Reliable Delivery in Wireless Sensor Networks*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 5, NO. 12, DECEMBER 2006.

[57] H. Kwon, T. Kim, S. Choi, B.G. Lee, F *A Cross-Layer Strategy for Energy-Efficient Reliable Delivery in Wireless Sensor Networks*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 5, NO. 12, DECEMBER 2006.

[58] L. Feeney, M. Nilson, *Investigating the energy consumption of a wireless network interface in an ad hoc networking environment*, IEEE INFOCOM 2001, Anchorage, Alaska, April 2001.

[59] IEEE, *IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Computer Society, LAN/MAN standards committee October 2003.

[60] IEEE, IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, LAN/MAN standards committee, October 2003.

[61] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, *Energy-efficient, collision-free medium access control for wireless sensor networks*, Sensys'03, Los Angeles, California November 2003.

[62] V. Rajendran, J.J. Garcia-Luna-Aceves, K. Obraczka, *Energy-efficient, application-aware medium access for sensor networks*, IEEE MASS 2005, Washington, November 2005.

[63] M. Garey, D. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[64] M. Conti, G. Maselli, G. Turi, S. Giordano, *Cross-layering in mobile ad hoc network design*, Computer, Vol.2, February 2004.

[65] D. Brelaz, *New methods to color the vertices of a graph*, Communications of the ACM, 22(4), 1979.

[66] J. Peemoller *A correction to Brelaz's modification of Brown's coloring al- gorithm*, Communications of the ACM, 26(8), 1983.

[67] M. Kubale, L. Kuszner, *A better practical algorithm for distributed graph coloring*, PARELEC 2002, Warsaw, Poland, September 2002.

[68] J. Hansen, M. Kubale, L. Kuszner, A. Nadolski, *Distributed largest-first algorithm for graph coloring*, EURO-PAR 2004, Dresden, Germany August 2004.

[69] A. Kosowski, L. Kuszner, *On greedy graph coloring in the distributed model*, Euro-Par 2006, Pisa, Italy, August 2006.

[70] T. Herman, S. Tixeuil, *A distributed TDMA slot assiognment algorithm for wireless sensor networks*, ALGOSENSORS 2004, Turku, Finland, July 2004.

[71] O. Johansson, *Simple distributed* $(\Delta + 1)$ *-coloring of graphs*, Information processing Letters, 70, 1999.

[72] I. Finoccho, A. Panconesi, R. Silvestri, *Experimental analysis of simple distributed vertex coloring algorithms*, SODA 2002, San francisco, California, January 2002.

[73] F. Kuhn, R. Watttenhofer, *On the complexity of distributed graph coloring*, PODC 2006, Denver, Colorado, July 2006.

[74] C. Busch, M. Magdon-Ismail, F. Sivrikaya, B. Yener, *Contention-free MAC protocols for wireless sensor networks*, DISC 2004, Amsterdam, Netherlands, October 2004.

[75] W. Kunvilaisah, Y. Hou, Q. Zhang, W. Zhu, C. Jay Kuo, Y. Zhang, *A cross layer quality of service mapping architecture for video delivery in wireless networks*, IEEE Journal on Selectead areas in Communications, Vol21, N10, December 2003.

[76] V. Kawadia, P. Kumar, *A cautionary perspective in cross layer design*, IEEE Wireless Communication, Vol12, February 2005.

[77] M. Schaar, S. Shankar, *Cross layer wireless multimedia transmission: challenges, principles and new paradigms*, IEEE Wireless Communication, Vol12, N4, August 2005.

[78] R. Winter, J. Schiller, N. Nikaeim, C. Bonnet, *Crosstalk: cross layer decision support based on global knowledge*, IEEE Communication Magazine, Vol44, January 2006.

[79] V. Raisinghani, S. Iyer, *ECLAIR: an efficient cross layer architecture for wireless protocol stacks*, 5th World Wireless Congress, San Francisco, USA May 2004.

[80] I. Rhee, A. Warrier, L. Xu, *Randomized dining philosophers to TDMA scheduling in wireless sensor networks*, Technical Report TR-2005-21, Dept of Computer Science, North Carolina State University, April 2005.

[81] J. Macker, I. Downard, J. Dean, B. Adamson, *Evaluation of Distributed Cover Set Algorithms in Mobile Ad Hoc Network For Simplified Multicast Forwarding*, ACM SIGMOBILE Mobile Computing and Communications Review, New York, NY, USA, July 2007.

[82] D. Durand, R. Jain, D. Tseytlin, *Distributed Scheduling Algorithms to Improve the Performance of Parallel Data Transfers*, Proc. IPPPS'94 Workshop on Input/Output in Parallel Computer Systems, April 1994.

[83] D. Durand, R. Jain, D. Tseytlin, *Applying Randomized Edge Coloring Algorithms to Distributed Communication: An Experimental Study.*, ACM Symposium of Parallel Algorithms and Architectures, 1995.

[84] International Telecommunication Union, *Method for point-to-area predictions for terrestrial services in the frequency range 30 MHz to 3000 MHz*, ITU-R P.1546-1, 2003.

[85] M. V. Marathe, A. Panconesi, L. D. Risinger, *An Experimental Study of a Simple, Distributed Edge-Coloring Algorithm*, ACM Journal of Experimental Algorithmics, Vol. 9, 2004.

[86] S. Gandham, M. Dawande, R. Prakash, *Link scheduling in sensor networks: distributed edge coloring revisited*, INFOCOM 2005, Miami, March 2005.

[87] H. Tamura, K. Watanabe, M. Sengoku and S. Shinoda, *On a New Edge Coloring Related To Multi-hop Wireless Networks*, APCCAS'02, Singapore, October. 2002.

[88] T.C. Barrett, G. Istrate, A. Kumar, M. Marathe, S. Thite, S. Thulasidasan, *Strong edge coloring for channel assignment in wireless radio networks*, IEEE International Workshop on Foundation and Algorithms for Wireless Networking, Pisa, Italy, March 2006.

[89] C.D. Young, *USAP: a unifying dynamic distributed multichannel TDMA slot assignment protocol*, IEEE MILCOM 1996, Vol. 1, October 1996.

[90] C.D. Young, *USAP multiple access: dynamic resource allocationfor mobile multihop multichannel wireless networking*, IEEE MILCOM 1999, Vol. 1, November 1999.

[91] J. Elson, L. Girod, D. Estrin, *Fine-Grained Network Time Synchronization using Reference Broadcasts*, OSDI 2002, Vol 36, pp. 147Ũ163. November 1999.

[92] S. Ganeriwal, R. Kumar, M. Srivastava, *Timing-Sync Protocol for Sensor Networks*, Proc. First Int. Conf. on Embedded Networked Sensor Systems, Los Angeles, California, November 2003.

[93] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, *The Flooding Time Synchronization Protocol*, SenSys 2004, Baltimore, Maryland November 2004

[94] S. Ganeriwal, R. Kumar, S. Adlakha, and M. Srivastava, *Network-wide Time Synchronization in Sensor Networks*, Technical Report, Networked and Embedded Systems Lab, Elec. Eng. Dept., UCLA, 2003.

[95] ZigBee Alliance, *ZigBee Specification*, http://www.zigbee.org/, December 2006.

[96] S. Gobriel, D. Mosse, R. Cleric, *TDMA-ASAP: senssor network TDMA scheduling with adaptive slot stealing and parallelism*, ICDCS 2009, Montreal, Canada, June 2009.

[97] I. Rhee, A. Warrier, M. Aia, J. Min, *Z-MAC: a hybrid MAC for wireless sensor networks*, SenSys'05, San Diego, California, November 2005.

[98] J-Y. Le Boudec and P. Thiran, *A Theory of Deterministic Queuing Systems for the Internet*, In Lecture Notes in Computer Science (LNCS), Vol. 2050, May 2004.

[99] M. Bertin, A. Bossche, G. Chalhoub, T. Dang, S. Mahfoudh, J. Rahmé, J. Viollet *OCARI for industrial wireless sensor networks*, IFIP Wireless Days 2008, Dubai, United Arab Emirates, November 2008.

[100] S. Mahfoudh, P. Minet, *Le format des message reseau*, Livrable ANR, Juin 2009.

[101] P.Minet, S. Mahfoudh, *Primitives utilisées par la couche réseau*, Livrable ANR, Juin 2009.

[102] A. Koubaa, M. Alves, E. Tovar, *Modeling and worst-case dimensioning of cluster-tree wireless sensor networks*, RTSS 2006, Rio di Janeiro, Brazil, December 2006.

[103] A. Koubaa, M. Alves, E. Tovar, *GTS allocation analysis in IEEE 80.15.4 for real-time wireless sensor networks* WPDRTS 2006, Rhodes Island, Greece, April 2006.

[104] A. Koubaa , M. Alves, E. Tovar, *Modeling and worst-case dimensioning of cluster-tree wireless sensor networks: proofs and computation details*, technical report IPP-HURRAY, TR-060601, http://www.dei.isep.ipp.pt/ãkoubaa/submissions.htm, 2006.

[105] D. Nguyen, P. Minet, *Scalability of the OLSR protocol with the Fish Eye extension*, ICN 2007, IEEE International Conference on Networking, Sainte-Luce, Martinique, April 2007.

[106] C. Adjih, P. Minet, T. Plesse, A. Laouiti, A. Plakoo, M. Badel, P. Muhlethaler, P. Jacquet, J. Lecomte, *Experiments with OLSR routing in a MANET*, Information Systems Technology NATO Symposium, Rome, Italy, April 2005.NATO, 2005.

[107] G. Chalhoub, A. Guitton, M. Misson, *MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part A: MaCARI: a synchronized tree-based MAC protocol*, WSAN 2008, IFIP International Conference on Wireless Sensor and Actor Networks.

[108] E. Livolant, A. van den Bossche, T. Val, *MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part B: Optimization of the intra-star exchanges for MaCARI*, WSAN 2008, IFIP International Conference on Wireless Sensor and Actor Networks.

[109] NS-2 simulator, http://www.isi.edu/nsnam/ns

[110] HART Communication Foundation, *HART Protocol Revision 7.0*, August 2007.

[111] $OCARI^{TM}$, Consortium Web site: http://ocari.lri.fr

[112] C. Perkins, E. Belding-Royer, S. Das, *Ad hoc On demand Distance Vector (AODV) Routing*, IETF RFC 3561, 2003.

Abstract

In this thesis, we consider wireless ad hoc and sensor networks where energy matters. Indeed, sensor nodes are characterized by a small size, a low cost, an advanced communication technology, but also a limited amount of energy. This energy can be very expensive, difficult or even impossible to renew. Energy efficient strategies are required in such networks to maximize network lifetime. We distinguish four categories of strategies: 1. Energy efficient routing, 2. Node activity scheduling, 3. Topology control by tuning node transmission power and 4. Reduction of the volume of information transferred. Our contribution deals with energy efficient routing and node activity scheduling.

For energy efficient routing, the idea consists in reducing the energy spent in the transmission of a packet from its source to its destination, while avoiding nodes with low residual energy. The solution we propose, called EOLSR, is based on the link state OLSR routing protocol. We show by simulation that this solution outperforms the solution that selects routes minimizing the end-to-end energy consumption, as well as the solution that builds routes based on node residual energy. We then show how we can improve the benefit of energy efficient routing using cross layering. Information provided by the MAC layer improves the reactivity of the routing protocol and the robustness of routes. Moreover, taking into account the specificities of some applications like data gathering allows the routing protocol to reduce its overhead by maintaining routes only to the sink nodes.

Concerning node activity scheduling, since the sleep state is the least power consuming state, our aim is to schedule node state between sleeping and active to minimize energy consumption while ensuring network and application functionalities. We propose a solution, called SERENA, based on node coloring. The idea is to assign a color to each node, while using a small number of colors and ensuring that two nodes with the same color can transmit without interfering. This color is mapped into a slot in which the node can transmit its messages. Consequently, each node is awake during its slot and the slots granted to its one-hop neighbors. It sleeps the remaining time. We show how this algorithm can adapt to different application requirements: broadcast, immediate acknowledgement of unicast transmissions... The impact of each additional requirement is evaluated by simulation. An originality of this work lies in taking into account real wireless propagation conditions. Color conflicts are then possible. A cross-layering approach with the MAC layer is used to solve these conflicts. We also show how cross-layering with the application layer can improve the coloring performance for data gathering applications.

This work has been done for the ANR OCARI project whose aim is to design and implement a wireless sensor network for applications in harsh environments such as power plants and warships. The network layer including SERENA and EOLSR has been specified and is now under implementation.